

California State University, San Bernardino

**CSUSB ScholarWorks**

---

Theses Digitization Project

John M. Pfau Library

---

2005

## Chinese Restaurant Online System

Yaming Chen

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Software Engineering Commons](#)

---

### Recommended Citation

Chen, Yaming, "Chinese Restaurant Online System" (2005). *Theses Digitization Project*. 2833.  
<https://scholarworks.lib.csusb.edu/etd-project/2833>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

CHINESE RESTAURANT ONLINE SYSTEM

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Computer Science

---

by  
Yaming Chen  
June 2005

CHINESE RESTAURANT ONLINE SYSTEM


---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino


---

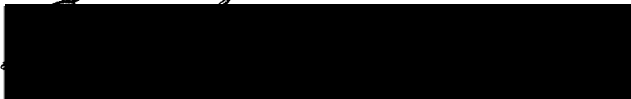
by  
Yaming Chen  
June 2005

Approved by:

  
\_\_\_\_\_  
Dr. Keith Schubert, Chair, Computer Science

April 13, 2005  
Date

  
\_\_\_\_\_  
Dr. Yasha Karant

  
\_\_\_\_\_  
Dr. David Turner

## ACKNOWLEDGEMENTS

This project was carried out under the supervision and guidance of Dr. Keith Schubert, from the Department of Computer Science, California State University, San Bernardino. I wish to thank Dr. Keith Schubert, my advisor, for his encouragement, support and guidance all along; Dr. Yasha Karant and Dr. David Turner, committee members of my study, for their valuable suggestions; Ms. Monica Gonzales, Mr. Kwon Soo Han, and Mr. Nam Kim from the Department, for their assistance and helps over the years.

## ABSTRACT

"Chinese Restaurant Online System" (CROS) is a piece of Business to consumer (B2C) e-commerce software, which means it can help businesses to sell the goods to the general public over the Internet. Additionally, the CROS can assist a Chinese restaurant to manage its business over the Web. There are two types of users of the system: the various customers and the staff of the Chinese restaurant. For the customers, with the CROS, they can view information about the restaurant over the Web and order some dishes online after they login to the system as well. For the restaurant, the employees can login to do their duties depending on their sponsored rights. Therefore, the CROS is not only a B2C e-commerce application, but also a kind of business management software.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	iii
ABSTRACT .....	iv
LIST OF TABLES .....	ix
LIST OF FIGURES .....	xi
CHAPTER ONE: INTRODUCTION	
1.1 Purpose of This Project .....	1
1.2 Scope of This Project .....	2
CHAPTER TWO: OVERALL DESCRIPTION	
2.1 Product Perspective .....	5
2.1.1 System Interfaces .....	5
2.1.2 User Interfaces .....	6
2.1.2.1 Home Page .....	6
2.1.2.2 Login Page .....	9
2.1.2.3 Register Page .....	10
2.1.2.4 Menu Page .....	11
2.1.2.5 Tray Page .....	12
2.1.2.6 Member Grid Page .....	14
2.1.2.7 Dish Detail Page .....	15
2.1.2.8 Customer Order Record Page ...	16
2.1.2.9 Make Schedule Record Page ....	17
2.1.2.10 Analysis Page .....	18
2.1.2.11 Customer Information Page ....	19
2.1.2.12 Make Schedule Page .....	20

2.1.3 Hardware Interfaces .....	22
2.1.4 Communication Interfaces .....	22
2.1.5 Database Interfaces .....	22
2.1.6 Memory Constraints .....	22
2.1.7 Operations .....	22
2.1.8 Site Adaptation Requirement .....	23
2.2 Product Functions .....	23
2.3 User Characteristics .....	25
2.3.1 Customer .....	25
2.3.2 Staff .....	25
2.3.3 Cashier or Operator .....	25
2.3.4 Manager .....	26
2.3.5 Administrator .....	26
2.4 Assumptions and Dependencies .....	26
2.5 Design Constraints .....	27
2.5.1 Standards Compliance .....	27
2.6 Software System Attributes .....	27
2.6.1 Reliability .....	27
2.6.2 Availability .....	28
CHAPTER THREE: DATABASE DESIGN	
3.1 Requirement Collections and Analysis .....	29
3.2 Database Schema Conceptual Model -Entity-Relationship Diagram .....	30
3.3 Choice of a Database Management System .....	30

3.4 Database Schema Logical Model -Relational Schema .....	32
3.5 Data Type and Details .....	33
CHAPTER FOUR: SYSTEM VALIDATION	
4.1 Unit Testing .....	43
4.2 Subsystem Testing .....	52
4.3 System Testing .....	54
CHAPTER FIVE: MAINTENANCE MANUAL	
5.1 Software Installation .....	55
5.1.1 Red Hat Installation .....	55
5.1.2 Install Apache .....	56
5.1.3 Install My Structured Query Language .....	57
5.1.4 Install Personal Home Page .....	58
5.1.5 Install Open Database Connectivity ...	59
5.2 Backup and Restore .....	60
5.2.1 System Backup .....	60
5.2.2 Database Backup .....	60
5.2.3 System Restore .....	60
5.2.4 Database Restore .....	61
CHAPTER SIX: CONCLUSION AND FUTURE DIRECTIONS	
6.1 Conclusion .....	62
6.2 Future Directions .....	63
APPENDIX A: CHINESE RESTAURANT ONLINE SYSTEM	
FILES PRINTOUT .....	64



REFERENCES .....	110
------------------	-----

## LIST OF TABLES

Table 1.	Structure of Table Card_types .....	33
Table 2.	Structure of Table Editorials .....	33
Table 3.	Structure of Table Editorial_categories ....	34
Table 4.	Structure of Table Categories .....	34
Table 5.	Structure of Table Familypackage .....	34
Table 6.	Structure of Table Family .....	35
Table 7.	Structure of Table Login .....	35
Table 8.	Structure of Table Info .....	35
Table 9.	Structure of Table Items .....	36
Table 10.	Structure of Table Reqs .....	36
Table 11.	Structure of Table Pay .....	37
Table 12.	Structure of Table Mem_level .....	37
Table 13.	Structure of Table Openhours .....	37
Table 14.	Structure of Table Members .....	38
Table 15.	Structure of Table Schedules .....	39
Table 16.	Structure of Table Orders .....	39
Table 17.	Structure of Table Size .....	40
Table 18.	Structure of Table State .....	40
Table 19.	Structure of Table Taketype .....	40
Table 20.	Structure of Table Type .....	41
Table 21.	Structure of Table Veg .....	41
Table 22.	Structure of Table Sauce .....	41
Table 23.	Structure of Table Tordern .....	42

Table 24. Structure of Table Rice .....	42
Table 25. Unit Test Results .....	43
Table 26. Unit Test Results (Class: Template) .....	51
Table 27. Unit Test Results (Class: DB_Sql) .....	52
Table 28. Subsystem Test Results .....	53
Table 29. System Test Results .....	54

## LIST OF FIGURES

Figure 1.	Deployment Diagram .....	6
Figure 2.	Home Page .....	9
Figure 3.	Login Page .....	10
Figure 4.	Register Page .....	11
Figure 5.	Menu Page .....	12
Figure 6.	Tray Page .....	14
Figure 7.	Member Grid Page .....	15
Figure 8.	Dish Detail Page .....	16
Figure 9.	Customer Order Record Page .....	17
Figure 10.	Make Schedule Record Page .....	18
Figure 11.	Analysis Page .....	19
Figure 12.	Customer Information Page .....	20
Figure 13.	Make Schedule Page .....	21
Figure 14.	Use Case Diagram for Users .....	24
Figure 15.	Entity-Relationship Diagram .....	31
Figure 16.	Database Relational Schema .....	32

## CHAPTER ONE

### INTRODUCTION

Conducting business activities online, in other words, participating in electronic commerce, is known as e-commerce. Today, there are three types of e-commerce: business to consumer, consumer to consumer, and business to business. Business to consumer (B2C) e-commerce consists of the sale of goods to the general public. Any product or service you purchase from a traditional retail store can be purchased on the Internet. For instance, instead of visiting a department store to purchase a DVD player, one can order it directly from the manufacturer's Web site with B2C e-commerce. Transactions can occur instantaneously and globally, therefore B2C e-commerce provides convenience for both the business and the consumer. This project proposed an example how to develop and implement a piece of Web application for a Chinese Restaurant Online System (CROS).

#### 1.1 Purpose of This Project

A lot of specific online applications have been developed and used in almost every field and the businesses have benefited from the use of these applications; however there is almost no specific Web application for Chinese

Restaurants. The purpose of this project is to develop a Chinese Restaurant Online System that can help various Chinese restaurants to extend their businesses. This system will be used by the different users, such as prospective customer, already registered customer, the restaurant chef or waiter, the cashier or operator, the manager, and the system administrator.

## 1.2 Scope of This Project

This software product will be shortly called CROS, and it will be an online application that caters to two different categories of customers:

- A. Prospective customer
- B. Already registered customer

It also caters to the other users:

- A. Chinese Restaurant Waiter and Chef
- B. Chinese Restaurant Cashier or Operator
- C. Chinese Restaurant Manager
- D. Chinese Restaurant Administrator

Each user will be issued various levels of privileges depending on his or her role. The CROS will attempt to accomplish the following functions:

1. Provide information for future customers who inquire on the Chinese restaurant menu, and order

dishes online based on the menu choices.

2. Provide information for already registered customer on the Chinese restaurant's menu, and if they are registered to allow them to order dishes online based on the menu choices.
3. Provide the general employee, such as waiter, waitress or chef, the ability to check their schedules and update their personal information.
4. Provide the restaurant cashier or operator a quick and efficient method to process and deal with the customers' orders in three categories: walk-ins, telephone, and online. Additionally, the cashier will generate the daily sale report that will be checked by the manager.
5. Provide manager a quick and efficient overview of total income in one day, and analyze the details of sale in the daily, monthly, or yearly, the number of orders, and so on. Also, the manager makes the working schedule for the employees.
6. Provide the system administrator access for maintenance of system. He has a high priority right so he can manipulate any kind of data in database, such as modify data, delete data, or add data into the database.

The next chapter will discuss about an overall description of the CROS.



## CHAPTER TWO

### OVERALL DESCRIPTION

#### 2.1 Product Perspective

Open source has brought a lot more than Linux to the computing world. It has also given us Personal Home Page (PHP) and My Structured Query Language (MySQL). According to Grademe, PHP and MySQL are the world's best combination for creating data-driven sites. Therefore, the CROS product will be a combination application of PHP and MySQL.

##### 2.1.1 System Interfaces

The Chinese restaurant online system is a 3-tier architecture. The first tier is the Client that displays user interfaces in a Web browser. The middle one is the Web Server running on the Linux Operating System. It uses Apache, which supports PHP, and the CROS resides here. The third tier is the Database Server using MySQL for a relational database management system (RDMS). The CROS database is stored here. The Web Server communicates with the Client using HTTP and with the Database Server via Open Data Base Connectivity (ODBC).

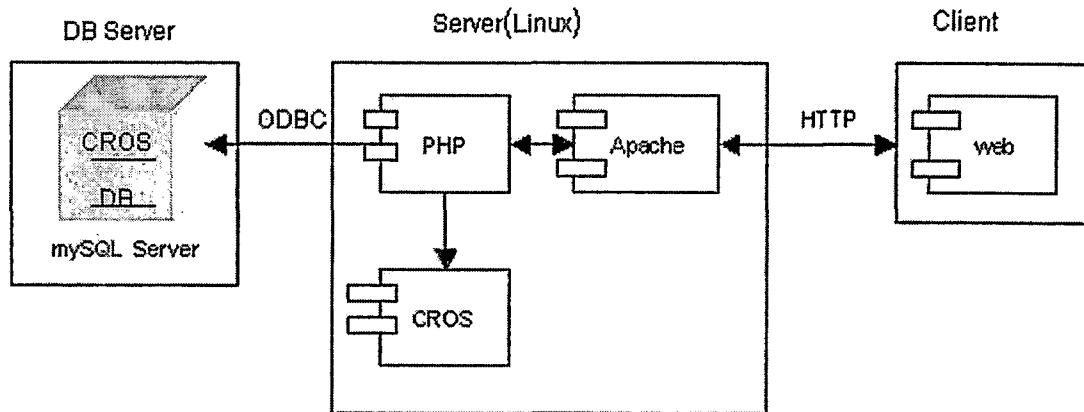


Figure 1. Deployment Diagram

### 2.1.2 User Interfaces

All user interfaces will be interactive and dynamic. They will allow the users to enter data into the database, or retrieve data from the database for viewing. The user login will determine, based on predetermined privileges, what data the user is allowed to enter and view. Future compatibility with web browser plug-ins can also allow adding additional dynamic and static interfaces such as viewing PDF documents (e.g., Adobe Acrobat Reader plug-in for web browsers).

Each type of user will have different options available to her or him. However the user interface must have a consistent look and feel. The necessary pages are listed in the following section.

#### 2.1.2.1 Home Page. Figure 2 shows the homepage of the

CROS. The page from top to bottom falls into three parts: the header, the content, and the footer.

The header section contains two-row items. In the left side of the first row there is a restaurant logo button. Clicking this button will take a user to the CROS' homepage (which is initially set to the CROS Home Page). The right side of this row is an ad element, which displays dynamically a friendly welcome slogan. Immediately below the first row, there are five navigational buttons that enable a user to access five corresponding pages. The functions of these buttons are briefly described here. The Home button is used for entering the current page from any other pages; the Menu is an access to the restaurant's Menu Page; the Sign In/Off button is for Login; Sign UP is designed for a registration, if one would like to become a CROS' member; and the last button, Tray, allows a user to make a final check on his or her on-line order.

Follow the header section is the content part. It provides a brief overview of the Chinese restaurant to a visitor. In this section, a user can find the specific information in which he or she is interested. The Search panel is designed for a costumer to choose the food items. To do this, one can either fill in the "dish" box with a desirable item, or just click the Search button; the

browser will scroll a list of dishes in the Sub Menu Page. In addition, the Advanced Search hyperlink provides more inquiring options to a costumer. The Categories panel shows a list of categories. A costumer can click the hyperlink of a specific category name to review the dish shown in the Sub Menu Page. In the Information panel, a user can find the restaurant's address, phone and location with a map. The enlarged map can be viewed by directly clicking on the map. To view clearly the map, one can click directly on the map to get an enlarged map. Alternatively, one can get the driving direction by clicking the hyperlink of 'Map' that links to the Yahoo Driving Guide Page. The "OPEN DAYS" panel shows working hour information. In the "Favorite" panel, some favorite dishes are displayed. If a visitor wants to order one of them, he or she can click a specific dish name. In this case, the user has to login to the system first. A visitor may also enjoy the enlarged picture of the dishes by clicking those images. The "New or Recommendation" panel provides the similar features with the "Favorite" ones.

The footer is located on the bottom area of the page. It includes two-row items. The first row includes five hyperlinks. Their features are the same as the above five-button ones in the header section. In the second row there

are the copyright information and the us hyperlink that allows a user to access the Outlook Express email software. This feature enables a user send some comments to the Webmaster.

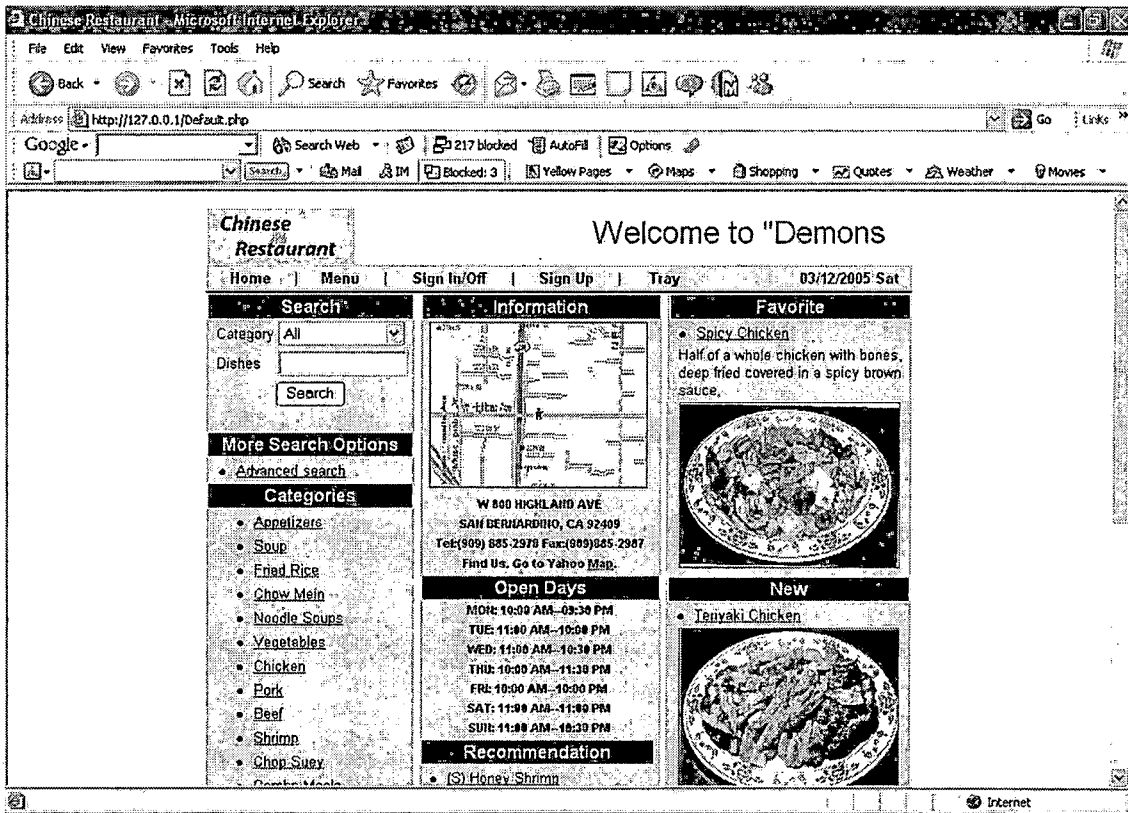


Figure 2. Home Page

2.1.2.2 Login Page. Figure 3 is the Login Page. This page is displayed when a user clicks the Sign In/Off button in the Home Page. The user's login information determines a type of a user. A new user does not need to be part of the system. The information required is the user's ID and

Password. The Sign In button will log a user into the system.

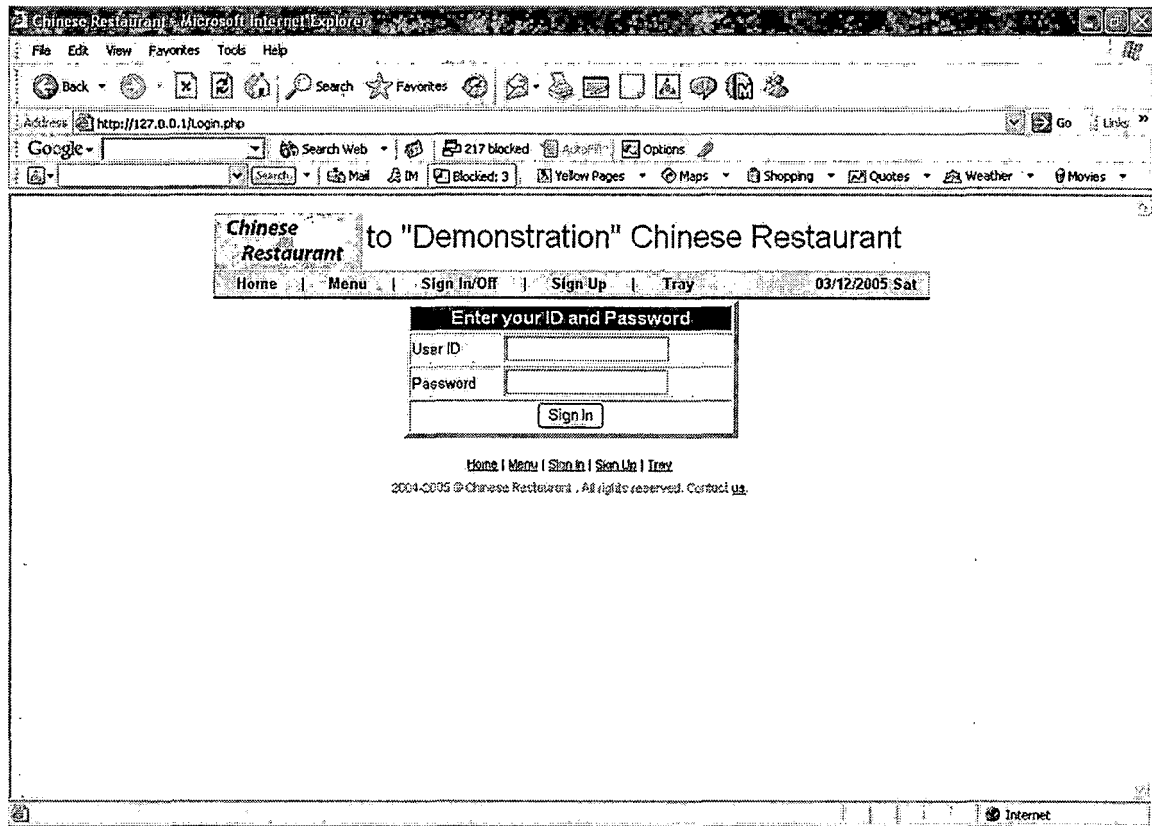


Figure 3. Login Page

2.1.2.3 Register Page. Figure 4 is the Register Page. It will show up when a user clicks the Sign Up button in the Home Page. This page provides two information fields that need to be filled for a new user who wants to become a member of the system. The one with asterisk is required. The other is optional. The Register button will insert the user's personal information into database and then take the

user back to the Home page, while the Cancel button will just take the user back to the Home Page.

The screenshot shows a web browser window titled "Chinese Restaurant" with the address bar displaying "http://127.0.0.1/Registration.php". The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The address bar also shows a Google search bar and various icons for search, mail, and other services. The main content area displays the "Chinese Restaurant" logo and a navigation menu with links for Home, Menu, Sign In/Off, Sign Up, and Tray. The date "03/12/2005 Sat" is shown in the top right corner. The registration form is titled "Registration" and contains the following fields: Login, Password, Confirm Password, First Name, Last Name, Company, Phone, Address, City, State (a dropdown menu), Zip, Type (a dropdown menu with "Online" selected), Take Type (a dropdown menu with "Delivery" selected), Pay Type (a dropdown menu with "Cash" selected), and Credit Card Type (a dropdown menu). The form is set against a light gray background with a subtle pattern.

Figure 4. Register Page

2.1.2.4 Menu Page. Figure 5 is the Menu Page. It is displayed to a user when clicking on the Menu button from the Home Page. This page displays a list of all the dishes provided by the restaurant. All these dishes are hyperlinked to the Dish Detail Page. For a user who has already logged into the system, he or she can click a specific dish name and order it. For a user who has not

signed in, he or she can view a description of a dish by clicking a specific dish image that is hyperlinked to the Dish Image Page.



Figure 5. Menu Page

2.1.2.5 Tray Page. This page appears right after a member user has just signed in. It presents the user's personal information and his or her order list if it is available. In the User Information section, the Login ID is linked via hyperlink to the My Info Page, which allows a user modify his or her personal information. In the Order



section, it will provide a list of the dishes that he or she has just ordered at this time. If it is available, a user can click the specific Edit hyperlink to modify this dish in the Detail Page. The Confirm button allows a user to confirm his or her order. The Last Order hyperlink links to the Last Order Page that allows the user to check last order if available. In addition, for an employee user from the restaurant, the some action buttons will show up below the header section. The number of the buttons will depend on the employee's role. With them, the restaurant employee can perform his or her duties.

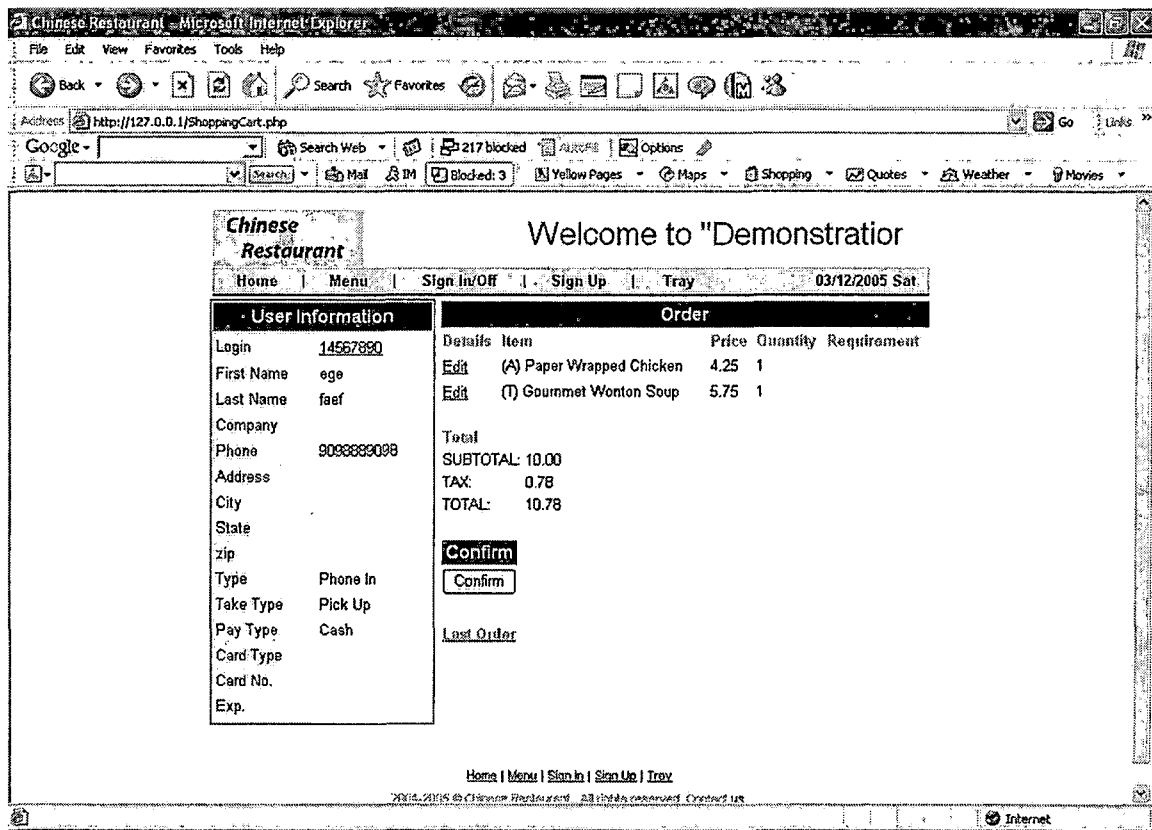


Figure 6. Tray Page

2.1.2.6 Member Grid Page. This page is displayed for the manager or administrator when he or she login to the system by clicking Member button. It provides a full member record list for the user. In the Search section, the user can quickly find some member records by entering data into the text box and then clicking the Search button. In the Member section, the user can click a column title hyperlink to get a sorted member record list. Additionally, the hyperlink Insert will link to the Member Record Page if the user wants to add more members into the database.

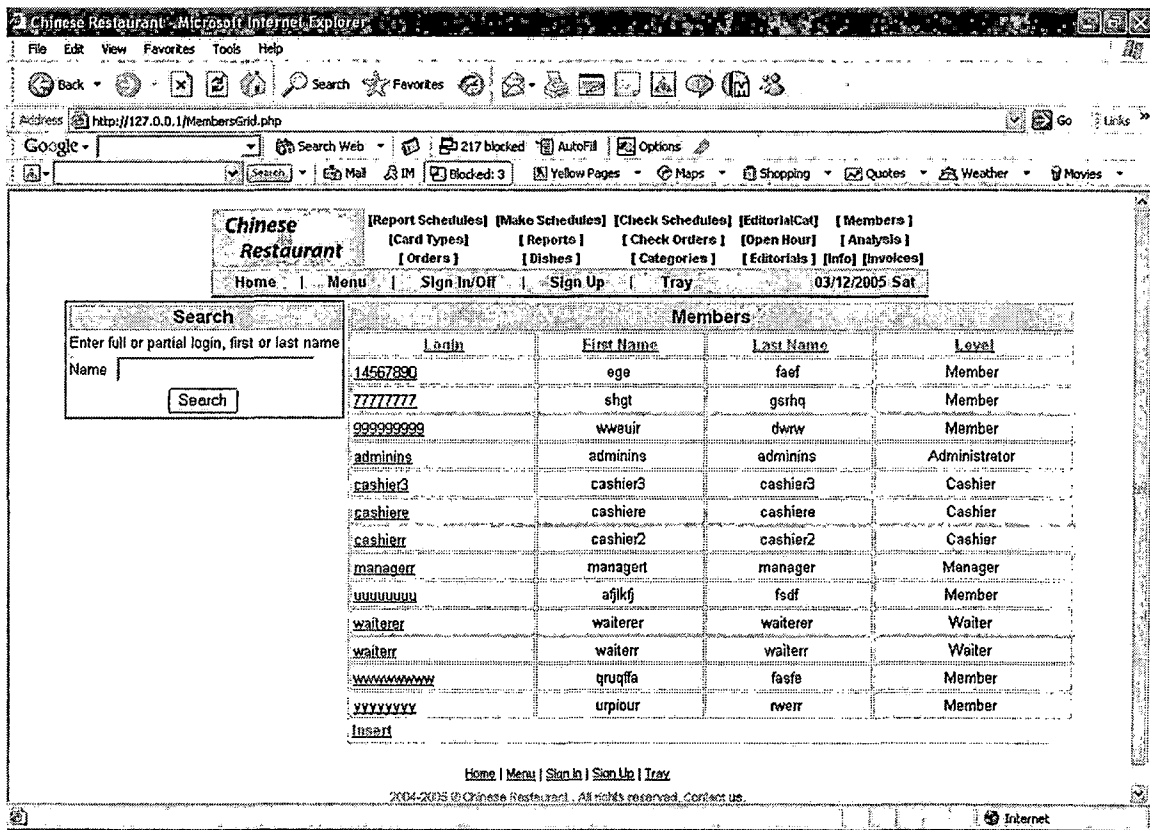


Figure 7. Member Grid Page

2.1.1.7 Dish Detail Page. This page will display when customer who has logged into the system clicks a specific Dish Name hyperlink in the Menu Page. It provides the dish detailed description. The customer can enter correlated data and then click Adding Dish on Tray button to finish his or her order. This button will take him or her to the Tray Page.

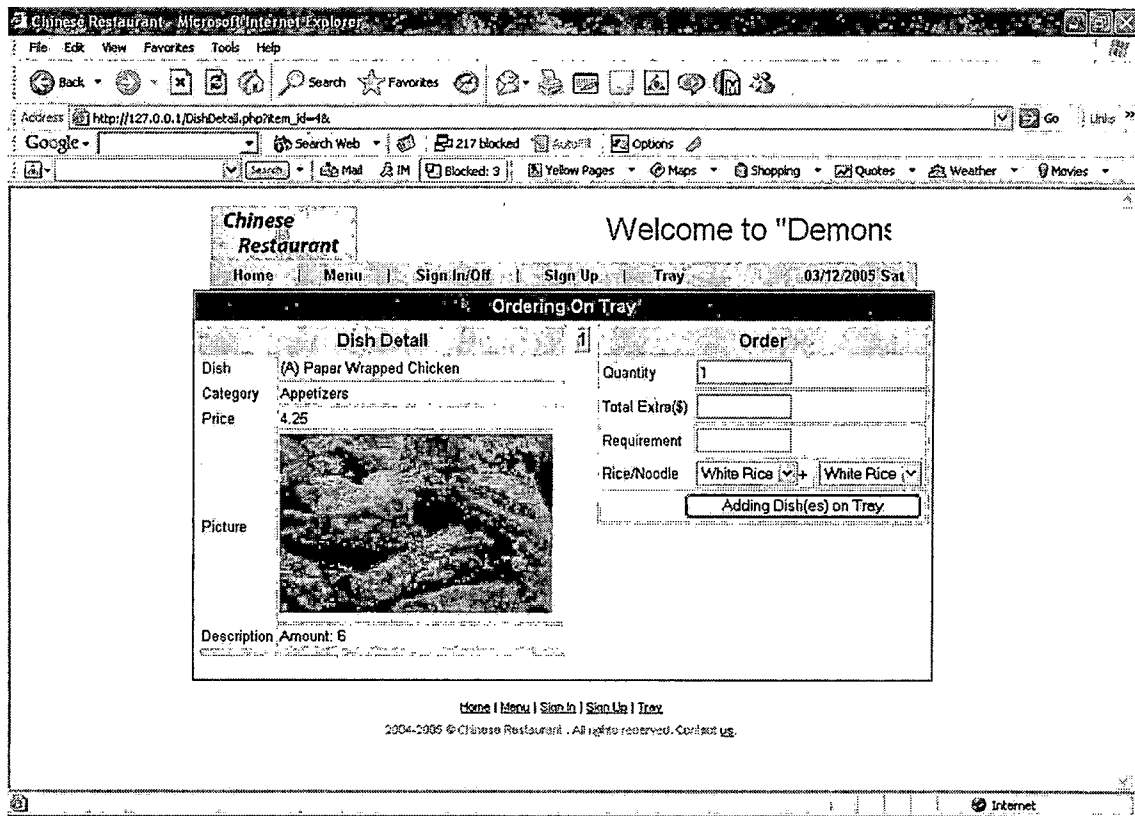


Figure 8. Dish Detail Page

2.1.1.8 Customer Order Record Page. This page will allow the cashier to help the customer to order his or her dishes.

Chinese Restaurant - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Print Mail News RSS Feeds

Address http://127.0.0.1/ChineseRestaurantRecord.php?order\_id=176 Go

Google Search Web 217 blocked AutoFill Options

Search Mail IM Blocked: 9 Yellow Pages Maps Shopping Quotes Weather Movies

---

**Chinese Restaurant**

[Report Schedules] [Make Schedules] [Check Schedules]

[Reports] [Check Orders]

Home Menu Sign In/Out Sign Up Tray 03/12/2005 Sat

Order for Customer ID:

FAM	Item	QTY	Rice/Noodle	Sauce(size/sauce)	AddVeg/Meat(size/veg)
(M)	Beef Chow Mein	1	White Rice	White Rice	
(G)	Cashew Chicken	2	White Rice	White Rice	
			White Rice	White Rice	
			White Rice	White Rice	
			White Rice	White Rice	
			White Rice	White Rice	

Insert Cancel

Home | Menu | Sign In | Sign Up | Tray

2004-2005 © Chinese Restaurant . All rights reserved. Contact us.

Done Internet

Figure 9. Customer Order Record Page

2.1.1.9 Make Schedule Record Page. This page will allow the manager to make the working schedule for the staff.



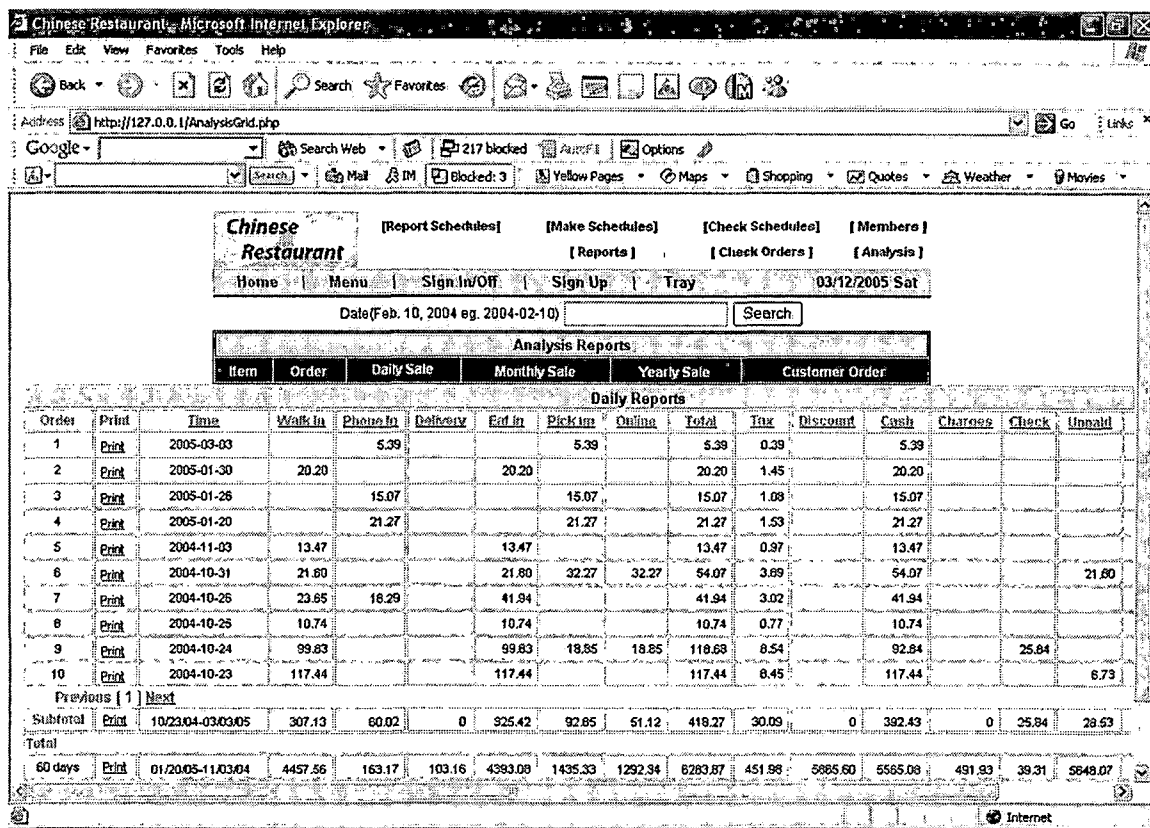


Figure 11. Analysis Page

2.1.1.11 Customer Information Page. This page will allow the cashier or operator to check the specific customer's order summary and status. It is displayed for the cashier when he or she clicks an Invoice ID hyperlink in the Check Order Page. It provides the order information based on the data provided by the customer or cashier. The cashier clicks the customer ID to link his or her Customer Record Page to modify the customer data. The cashier can click the Confirm Order to link the Confirm Record Page in order to finish his customer's order. The hyperlink Add





schedule if it exists. The hyperlink Edit will show up if the user is manager. He or she can go to Schedule Record Page in order to modify schedule for his or her worker by clicking the Edit hyperlink. The manager can also click on hyperlink Add New to go to Schedule Record Page. When the employee is chef, waiter, waitress, or cashier, or other worker, Edit and Add New hyperlinks will not appear because he or she has not right to modify this data.

**Chinese Restaurant** [Report Schedules] [Make Schedules] [Check Schedules]

Home | Menu | Sign In/Out | Sign Up | Tray | 03/12/2005 Sat

**Search**  
 Name:  All  
 Date:

March 2005 April 2005

**Attention**  
The working schedule until 2005-03-19

[Previous Week](#) [Next Week](#)

**The Working Schedule 03/06/05--03/12/05**

Name	ID	03/06 Sun	03/07 Mon	03/08 Tue	03/09 Wed	03/10 Thu	03/11 Fri	03/12 Sat	Days
admins	1		02:00PM-08:00PM						1
cashiers	3			11:00AM-06:30PM					1
manager	6				02:30PM-08:30PM				1
waiters	5			11:00AM-07:00PM 07:30PM-08:30PM			11:30AM-04:00PM		2
<b>Total Workers</b>		0	1	2	1	0	1	0	5

Home | Menu | Sign In | Sign Up | Tray

Figure 13. Make Schedule Page

### 2.1.3 Hardware Interfaces

CROS will not need any specific hardware interface. The only requirement is that a networking card has to be connected to the Internet to obtain the information. The hardware interface should be supported by any operating system.

### 2.1.4 Communication Interfaces

The HTTP is used as the communication interface for Internet. In addition, ODBC is used as the other communication interface between database and the software product.

### 2.1.5 Database Interfaces

MySQL database will be added for the server system. Once the database has been created, interfaces can access the database through PHP using ODBC. Information got will be shown out using PHP and HTML tags.

### 2.1.6 Memory Constraints

CROS system would not need any new software installations. As a thumb rule, more memory would improve the performance.

### 2.1.7 Operations

CROS would be running on the web server. The whole system would be available anytime to all user categories. Once the data is updated to the system it would be

consistently available to any user using the system.

#### 2.1.8 Site Adaptation Requirement

No specific on-site requirement is necessary. We assume that the user would use any kind of Web browser that would be available to access the system.

### 2.2 Product Functions

Figure 14 shows the Use case diagrams that graphically depict the users and the principle functions of CROS.

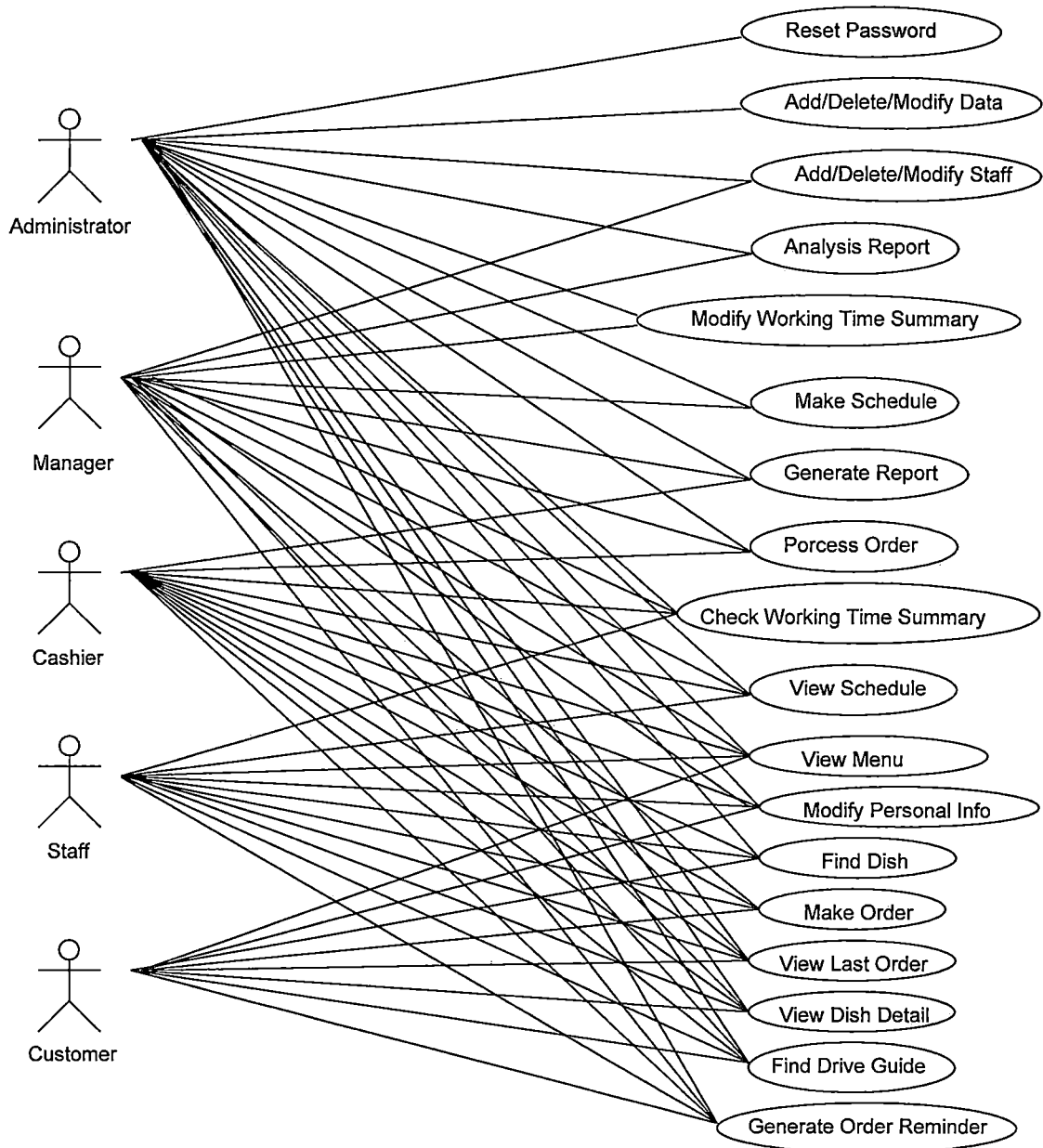


Figure 14. Use Case Diagram for Users

## 2.3 User Characteristics

The users of the CROS would fall into one of the following groups:

### 2.3.1 Customer

The users in this group are two kinds of customers including prospective customer and currently existing customer looking for information about the order. The user will not need a login to access the system. All the information would be provided in an interactive manner for the customer to plan out the order and to get an idea on order needed as their specific requirements. Also, the users in this group can order some dishes online according to their specific requirements.

### 2.3.2 Staff

The user in this group is the staff, such as waiter, waitress, and chef, etc, of the Chinese restaurant. The user will be able to login to the system and view working schedule, modify personal information.

### 2.3.3 Cashier or Operator

The user in this group is a cashier or operator of the Chinese restaurant. The user will be able to login to access the system and search order information, search order, add order, modify order by the customer requirements and generate a daily sale report. In addition, like the

staff, he or she can also view the work schedule and modify personal information.

#### 2.3.4 Manager

The user in this group is the manager of the Chinese restaurant. The user will be able to login to the system and make working schedule, check order, generate report, and modify all member personal information except administrator one.

#### 2.3.5 Administrator

The user in this group is the administrator of the CROS or any other employee appointed by the boss to perform the duties. The user will be able to login to the system and add data, delete data, modify data, and backup database.

### 2.4 Assumptions and Dependencies

In this environment, we assume there are no network connectivity problems. Furthermore, we assume there is no hardware resource limitation. We are assuming that a Web Browser such as Microsoft Internet Explorer, Opera, Konqueror, and Mozilla is available on the client system. Also, we assume no problems associated with the web server or the database server.

## 2.5 Design Constraints

CROS shall employ the following as a minimum:

- CROS shall have a GUI.
- CROS shall use client/server architecture.
- CROS shall use an internal database for storage of information.
- CROS shall employ software interfaces as lay out in section 2.1, external interfaces.

### 2.5.1 Standards Compliance

CROS will comply with:

- Customization for needs of a customer and staff.
- Rules and regulations of the Chinese restaurant.
- The information presented will be correct and usable by the potential users.

## 2.6 Software System Attributes

The CROS will run as an application on a web server and communicate with client's browser through the HTTP protocol.

### 2.6.1 Reliability

The CROS will be designed to be a highly robust online system. Errors will be trapped and appropriate messages will be displayed.

### 2.6.2 Availability

The CROS will be available as a published web address. The component will remain available at that address, except when the systems administrator takes the system down for maintenance purposes.



## CHAPTER THREE

### DATABASE DESIGN

Previous chapter of this project briefly described an overview of the CROS. This chapter will focus on database design. In order to accommodate the information needs of the users in an organization for a defined set of application, the database design falls five sequential steps phases: requirements collection and analysis, database schema conceptual model, choice of a Database Management System, database schema logical model, and data type and details. The following sections will describe them respectively.

#### 3.1 Requirement Collections and Analysis

The goals of database design are to satisfy the information content requirements of the specified users and applications. To meet the CROS requirements, the data collection depends on the properties of the CROS. Its data mainly consists of members, items, and orders. In the member data, it contains some necessary information, such as member\_id, creat\_time, pay\_type, member\_level, etc. For the item data, it needs some relevant information, such as item\_id, name, category\_id, etc. In the order data, it

includes order\_id, member\_id, item\_id, creat\_time, etc. The other data, in addition, is also necessary to meet the CROS requirements. For instance, editoiral\_categories, editorials, open hours, schedules, etc. Using php scripts will check all of the data when the associated user interface page is submitted to the server. CROS database schema will describe further in the following section.

### 3.2 Database Schema Conceptual Model -Entity-Relationship Diagram

Entity-Relationship Diagram can illustrate the entity types and relationship types by displaying their extensions. Figure 15 is an ER Diagram for CROS. It displays the individual entities and relationships instances.

### 3.3 Choice of a Database Management System

The choice of a Database Management System (DBMS) is governed by various factors, such technical, economic, etc. For the small business such as the Chinese restaurant, MySQL is a good choice. Because it is cheap, lightweight and fast database and it also supports PHP. These are great news for the small business, such as a small Chinese restaurant, developing small-scale client/server

applications that cannot afford to purchase one of the commercially available products.

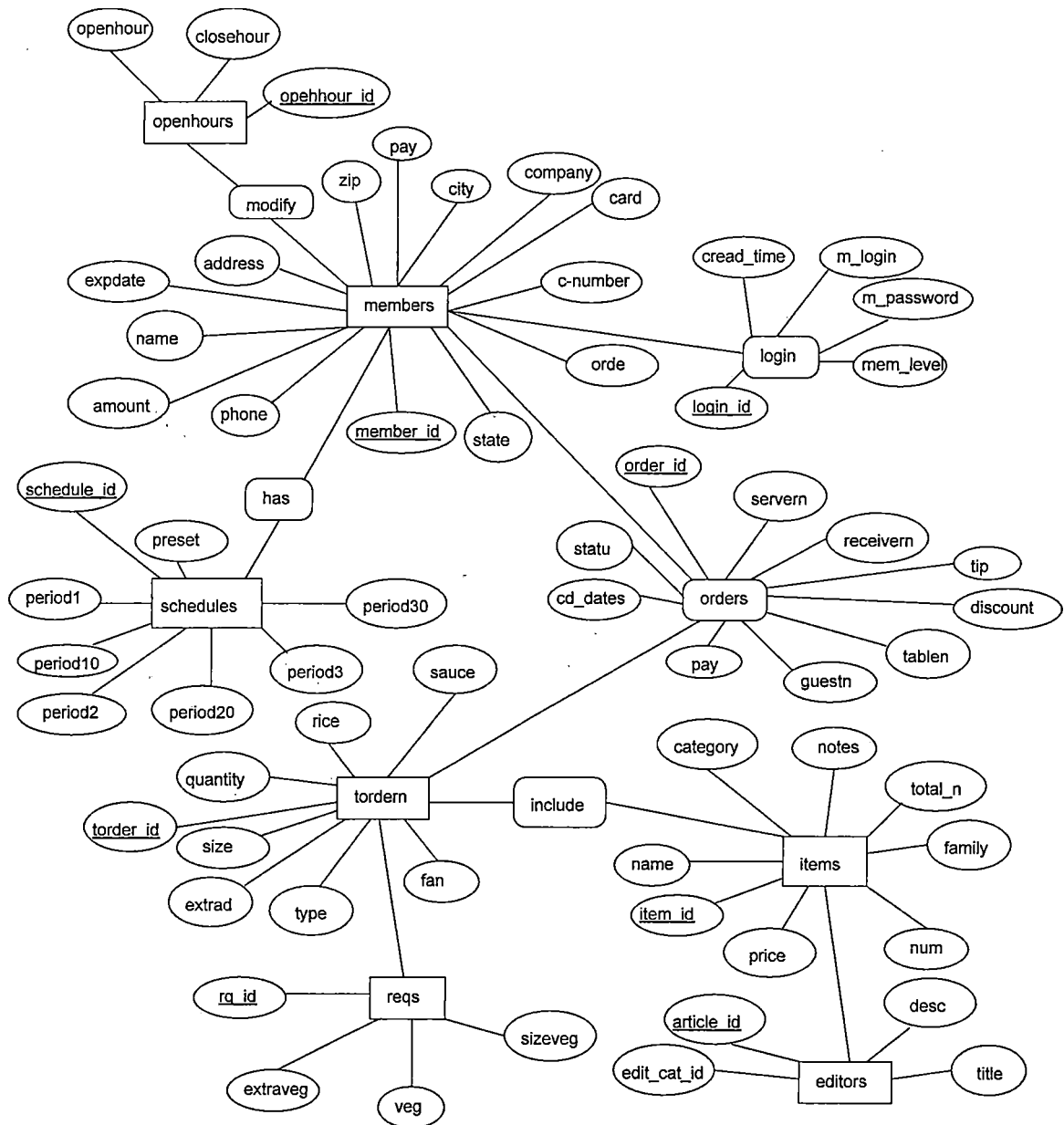


Figure 15. Entity-Relationship Diagram

### 3.4 Database Schema Logical Model - Relational Schema

Items					
<u>item_id</u>	name	category_id	price	family	hot_url
h s url	num	image url	total	notes	

Login				
<u>login_id</u>	<u>member_lo</u>	<u>member_pa</u>	<u>creat_time</u>	<u>member_le</u>

Members					
<u>member_id</u>	<u>first name</u>	<u>last name</u>	company	address	<u>state_id</u>
<u>card number</u>	<u>take type</u>	<u>creat-time</u>	<u>type_id</u>	<u>pay type</u>	<u>card type</u>
zip	expdate	phone	city	name	amount

Openhours		
<u>openhour_id</u>	<u>openhour</u>	<u>closehour</u>

Orders					
<u>order_id</u>	<u>member_id</u>	<u>statu</u>	week	pay	<u>cd_dates</u>
servern	receivern	tablen	questn	tip	discount

Reqs				
<u>rq_id</u>	<u>torder_id</u>	<u>extraveg</u>	<u>veg_id</u>	<u>sizeveg</u>

Schedules				
<u>schedule_id</u>	period2	<u>date_time</u>	period1	period10
<u>member_id</u>	period20	period3	period30	preset

Tordern				
<u>torder_id</u>	<u>item_id</u>	<u>order_id</u>	quantity	type
fan	<u>rice_id</u>	<u>rice id2</u>	<u>sauce_id</u>	<u>sizesauce</u>
extrad	requirement	<u>cd_times</u>		

Figure 16. Database Relational Schema

According to the Entity-Relationship Diagram above, all entities and their attributes are mapped into the relational tables. They are illustrated in figure 16.

### 3.5 Data Type and Details

The logical model establishes the following detailed design in MySQL database. The following tables describe data type, length, primary key, null, default, and extra information, such as auto\_increment (AI).

Table 1. Structure of Table Card\_types

Field	Type	Null	Key	Default	Extra
card_type_id	Tinyint(2)	Not	PRI		AI
Name	Varchar(50)	Not			

Table 2. Structure of Table Editorials

Field	Type	Null	Key	Default	Extra
article_id	Tinyint(2)	Not	PRI		AI
editor_cat_id	Tinyint(2)	Not		0	
article_title	Varchar(200)			Null	
article_desc	Text				

Table 3. Structure of Table Editorial\_categories

Field	Type	Null	Key	Default	Extra
editor_cat_id	Tinyint(2)	Not	PRI		AI
name	Varchar(50)	Not			

Table 4. Structure of Table Categories

Field	Type	Null	Key	Default	Extra
name	Varchar(50)	Not			
category_id	Tinyint(2)	Not	PRI		AI

Table 5. Structure of Table Familypackage

Field	Type	Null	Key	Default	Extra
family_id	Tinyint(1)	Not	PRI		AI
price	Float(8,4)	Not		0.0000	
type	Char(1)	Not		W	
numbe	Tinyint(1)	Not		0	
descri	Int(3)	Not			

Table 6. Structure of Table Family

Field	Type	Null	Key	Default	Extra
Fa_id	Tinyint(1)	Not	PRI		AI
name	Varchar(100)	Not			

Table 7. Structure of Table Login

Field	Type	Null	Key	Default	Extra
login_id	Int(10)	Not	PRI		AI
member_login	Varchar(20)	Not			
member_password	Varchar(20)	Not		0	
creat_time	Datetime	Not		0000-00-00 00:00:00	
member_level	Tinyint(1)	Not		1	

Table 8. Structure of Table Info

Field	Type	Null	Key	Default	Extra
info_id	Tinyint(3)	Not	PRI		AI
name	Varchar(50)	Not			

Table 9. Structure of Table Items

Field	Type	Null	Key	Default	Extra
item_id	Tinyint(3)	Not	PRI		AI
name	Varchar(50)	Not			
category_id	Tinyint(2)	Not		0	
price	Float(4,2)	Not		0.00	
image_url	Varchar(100)				
hot_url	Varchar(100)				
h_s_url	Varchar(100)				
num	Int(11)				
family	Tinyint(1)	Not		0	
total_n	Int(11)			0	

Table 10. Structure of Table Reqs

Field	Type	Null	Key	Default	Extra
rq_id	Int(11)	Not	PRI		AI
torder_id	Int(11)	Not		0	
extraveg	Float(4,1)			0.0	
veg_id	Tinyint(2)			0	
sizeveg	Tinyint(2)			0	



Table 11. Structure of Table Pay

Field	Type	Null	Key	Default	Extra
pay_type_id	Tinyint(2)	Not	PRI		AI
name	Varchar(15)	Not			

Table 12. Structure of Table Mem\_level

Field	Type	Null	Key	Default	Extra
member_level_id	Tinyint(1)	Not	PRI		AI
name	Varchar(15)	Not		Null	

Table 13. Structure of Table Openhours

Field	Type	Null	Key	Default	Extra
openhour_id	Tinyint(1)	Not	PRI		AI
openhour	Time	Not		00:00:00	
closehour	Time	Not		00:00:00	

Table 14. Structure of Table Members

Field	Type	Null	Key	Default	Extra
member_id	Int(10)	Not	PRI		AI
first_name	Varchar(50)				
last_name	Varchar(50)				
company	Varchar(50)				
address	Varchar(50)				
State_id	Tinyint(2)				
zip	Varchar(10)				
city	Int(11)				
creat_time	Datetime	Not		00-00-00 00:00:00	
type_id	Tinyint(2)	Not		0	
pay_type_id	Tinyint(2)	Not		0	
card_type_id	Tinyint(2)				
card_number	Int(16)				
expdate	Varchar(5)				
phone	Varchar(10)				
take_type_id	Tinyint(1)	Not		0	
name	Varchar(15)			customer	
amount	Float(6,2)				
orde	Int(11)				
work_n	Tinyint(3)	Not		0	

Table 15. Structure of Table Schedules

Field	Type	Null	Key	Default	Extra
schedule_id	Int(11)	Not	PRI		AI
member_id	Int(10)	Not			
date_time	Date			0000-00-00	
period1	Time	Not		00:00:00	
period10	Time	Not		00:00:00	
period2	Time	Not		00:00:00	
period20	Time	Not		00:00:00	
period3	Int(3)			0	
period30	Int(3)			0	
preset	Tinyint(1)	Not		0	

Table 16. Structure of Table Orders

Field	Type	Null	Key	Default	Extra
order_id	Int(11)	Not	PRI		AI
member_id	Int(10)	Not		0	
statu	Tinyint(1)	Not		0	
cd_dates	Datetime			0000-00-00 00:00:00	
weekdate	Tinyint(1)	Not		0	
servern	Tinyint(2)				
receivern	Tinyint(2)				
tablen	Char(3)				
guestn	Tinyint(3)				
tip	Float(6,2)				
discount	Float(6,2)				

Table 17. Structure of Table Size

Field	Type	Null	Key	Default	Extra
size_id	Tinyint(2)	Not	PRI		AI
name	Varchar(30)	Not			

Table 18. Structure of Table State

Field	Type	Null	Key	Default	Extra
state_id	Tinyint(2)	Not	PRI		AI
name	Char(2)	Not			

Table 19. Structure of Table Taketype

Field	Type	Null	Key	Default	Extra
take_type_id	Tinyint(1)	Not	PRI		AI
name	Varchar(10)	Not			

Table 20. Structure of Table Type

Field	Type	Null	Key	Default	Extra
take_id	Tinyint(2)	Not	PRI		AI
name	Varchar(15)	Not			

Table 21. Structure of Table Veg

Field	Type	Null	Key	Default	Extra
veg_id	Tinyint(2)	Not	PRI		AI
name	Varchar(30)	Not			

Table 22. Structure of Table Sauce

Field	Type	Null	Key	Default	Extra
sauce_id	Tinyint(2)	Not	PRI		AI
name	Varchar(30)	Not			

Table 23. Structure of Table Tordern

Field	Type	Null	Key	Default	Extra
torder_id	Int(11)	Not	PRI		AI
item_id	Int(11)	Not		0	
order_id	Int(11)	Not		0	
quantity	Tinyint (3)	Not		0	
type	Char(1)			W	
fan	Tinyint (3)	Not			
rice_id	Tinyint (1)				
rice_id2	Tinyint (1)				
sauce_id	Tinyint (2)				
sizesauce	Tinyint (2)				
extrad	Float (4,1)				
requiremen t	Text				
cd_times	Datetime	Not		0000-00-00 00:00:00	

Table 24. Structure of Table Rice

Field	Type	Null	Key	Default	Extra
rice_id	Tinyint(1)	Not	PRI		AI
name	Varchar(10)	Not		Null	

## CHAPTER FOUR

### SYSTEM VALIDATION

To run CROS properly, testing system is very important. This chapter will discuss the system validation by the following sections.

#### 4.1 Unit Testing

Unit test, the basic level of testing, is testing of individual hardware or software units or groups of related units. During the unit testing phase, individual components are tested to ensure that they operate correctly. They can be object, class, program, etc. The unit testing results of CROS are shown in Table 25.

Table 25. Unit Test Results

Forms	Test Performed	Results
Home Page	<ul style="list-style-type: none"><li>• Check if all the links work correctly.</li><li>• Check if all the buttons work properly.</li></ul>	Pass

Forms	Test Performed	Results
Login Page	<ul style="list-style-type: none"> <li>• Check if all the links work correctly.</li> <li>• Verify if the page can work correctly by inputting data.</li> <li>• Check that all the buttons work properly.</li> </ul>	Pass
Register Page	<ul style="list-style-type: none"> <li>• Check if all the buttons work properly.</li> <li>• Verify if the page can work properly by inputting data.</li> </ul>	Pass
Large Map Page	<ul style="list-style-type: none"> <li>• Check if all the links work correctly.</li> </ul>	Pass
Menu Page	<ul style="list-style-type: none"> <li>• Check if all the links work correctly.</li> </ul>	Pass
Tray page	<ul style="list-style-type: none"> <li>• Check if all the links work correctly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Member Grid Page	<ul style="list-style-type: none"> <li>• Check if all the links work correctly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Schedule Detail Page	<ul style="list-style-type: none"> <li>• Check if all the links work correctly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Las Week Analysis Order Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass



Forms	Tests Performed	Results
Sub Menu Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Check Orders Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Customer Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Customer Information Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Confirm Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Customer Order Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Report Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass

Forms	Tests Performed	Results
Report Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Make Schedule Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Schedule Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Check Schedule Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Confirm schedule Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Confirm Schedule Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass

Forms	Test Performed	Results
Report Schedule Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Monthly Sale Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Editorial Category Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Editorial Category Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Yearly Sale Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Customer Order Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass

Forms	Tests Performed	Results
Card Type Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Analysis Customer Order Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Card Type Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Card Type Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Admin Dishes Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Dish Maintain Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass

Forms	Tests Performed	Results
Category Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Category Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Editorial Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Editorial Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Open Hour Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Open Hour Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Customer Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass

Forms	Tests Performed	Results
Customer Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Information Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> </ul>	Pass
Information Record Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Item Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass
Analysis Order Grid Page	<ul style="list-style-type: none"> <li>• Check if all links work properly.</li> <li>• Check if all the buttons work properly.</li> <li>• Verify handling valid data input.</li> </ul>	Pass

Table 26. Unit Test Results (Class: Template)

Functions	Tests Performed	Results
Template	<ul style="list-style-type: none"> <li>• Check if constructor works properly.</li> </ul>	Pass
NextDBlockName	<ul style="list-style-type: none"> <li>• Verify if next form is blocked properly.</li> </ul>	Pass
getBlock	<ul style="list-style-type: none"> <li>• Verify if get block properly.</li> </ul>	Pass
GetVar	<ul style="list-style-type: none"> <li>• Check if block name gets correctly.</li> </ul>	Pass
Set_var	<ul style="list-style-type: none"> <li>• Verify if block name is parsed correctly.</li> </ul>	Pass
Print_var	<ul style="list-style-type: none"> <li>• Verify if block name is printed correctly.</li> </ul>	Pass
Load_file	<ul style="list-style-type: none"> <li>• Check if the file is loaded correctly.</li> </ul>	Pass
PrintAll	<ul style="list-style-type: none"> <li>• Verify if the html files are printed correctly.</li> </ul>	Pass
ProceedTpl	<ul style="list-style-type: none"> <li>• Check if template proceeds properly.</li> </ul>	Pass
blockVars	<ul style="list-style-type: none"> <li>• Check if block variable returns correctly.</li> </ul>	Pass
replaceBlock	<ul style="list-style-type: none"> <li>• Verify if replace block correctly.</li> </ul>	Pass

Table 27. Unit Test Results (Class: DB\_Sql)

Functions	Tests Performed	Results
DB_Sql	<ul style="list-style-type: none"> <li>• Verify if constructor works properly</li> </ul>	Pass
connect	<ul style="list-style-type: none"> <li>• Check if connection works properly.</li> </ul>	Pass
query	<ul style="list-style-type: none"> <li>• Check if query get correct results.</li> </ul>	Pass
Next_record	<ul style="list-style-type: none"> <li>• Check if walk result set correctly.</li> </ul>	Pass
lock	<ul style="list-style-type: none"> <li>• Check table locking</li> </ul>	Pass
seek	<ul style="list-style-type: none"> <li>• Verify to find correct position in result set.</li> </ul>	Pass

## 4.2 Subsystem Testing

Subsystem testing is the next step in the testing process, where all related units from a subsystem do a certain task. Therefore, the subsystem test process is useful for detecting interface errors and specific functions. Table 28 shows subsystem test results in detail.



Table 28. Table Subsystem Test Results

Subsystem	Tests Performed	Results
Authorize subsystem	<ul style="list-style-type: none"> <li>• Test if subsystem can get the error message.</li> <li>• Make sure the result of authorizing user is correct.</li> <li>• Verify the login user information is store in session properly.</li> <li>• Check if the saving user login information function stores the user information.</li> <li>• Verify the login page redirect to the correct browsing request after the user logins in.</li> <li>• Check if users can update their own account properly.</li> </ul>	Pass
Data management subsystem	<ul style="list-style-type: none"> <li>• Make sure the subsystem checks the user before dealing with add, modify, or delete function.</li> <li>• Verify the subsystem check the user privilege before handling the user's date.</li> <li>• Verify if the subsystem show the page related to view user data is the users are the owner.</li> </ul>	Pass
Browsing subsystem	<ul style="list-style-type: none"> <li>• Check if the subsystem checks for user privilege before showing pages.</li> <li>• Verify the page is showing properly after the user click on the page line.</li> </ul>	Pass

### 4.3 System Testing

System testing is needed to ensure that every component of a system is operating as it should, and that the system is performing exactly in accordance with the specific local requirements. It is the highest level of testing. During the testing phase, the testing process uses real data to test the system. The data are what the system is intended to manipulate. The CROS system testing steps includes three steps: first integrating all subsystem into CROS; second starting up the entire server which CROS need to run such Apache server and MySQL database server, by using a variety of data runs the CROS to analyze the overall result. The CROS system testing steps as the following:

Table 29. System Test Results

System Testing	Results
Install CROS system into Web server	Pass
Start up Apache server and MySQL database server.	Pass
Running testing by using real data on all forms and reports.	Pass

## CHAPTER FIVE

### MAINTENANCE MANUAL

#### 5.1 Software Installation

In order to run CROS, the six packages will need to be installed properly in advance. They are Red Hat Linux 9, Apache, PHP, MySQL, and Open Database Connectivity (ODBC). The following sections will explain their installation steps respectively.

##### 5.1.1 Red Hat Installation

To run CROS, the first task is to choose an operating system. In this case, we pick Red Hat Linux 9.0 as our server platform. Its installation steps describe as following:

The hardware compatibility and disk space are particularly important before installation. Make sure a server installation require 850MB for a minimal installation.

Red Hat Linux 9.0 is distributed on 3 CDs. Start by booting the system with the first CD in the drive, making sure the bios is set to boot from CD before other devices.

Follow the install wizard and sets up the required information such as network setting and the hardware environment.

After configure and select individual the necessary packages, then install it, the machine will need to restart after Installation Redhat is done.

### 5.1.2 Install Apache

In order to install Apache, first log in as 'root'. After login, extract the sourceballs by typing the following commands:

```
#tar -zxf httpd-2.0.48.tar.gz
#tar -zxf mysql-4.0.16.tar.gz
#tar -zxf php-4.3.4.tar.gz
```

The commands above will extract the sourceballs into their own separate directories. Now lets move on to compiling the source into usable programs. We'll start with Apache.

Change into the directory created when you untarred the sourceball as follows:

```
#cd httpd-2.0.48
```

Follow this command by typing:

```
#!/configure -prefix=/usr/local/apache2 -enable-mods-  
shared=most (enter)  
#make  
#make install
```

Apache is installed with the exception of a few minor changes we still need to make. So issue the following

commands:

```
#vi /usr/local/apache2/conf/httpd.conf
LoadModule php4_module modules/libphp4.so
DirectoryIndex index.html index.html.var index.php
AddType application/x-httpd-php .php

Now we save the file and we are done with Apache.
```

### 5.1.3 Install My Structured Query Language

To install My Structured Query Language (MySQL), you also need to run the follow commands as 'root'. First change into the MySQL source directory as follows:

```
#cd mysql-4.0.16
```

Follow these commands by typing:

```
#./configure --prefix=/usr/local/mysql --
localstatedir=/usr/local/mysql/data --disable-maintainer-
mode --with-mysqld-user=mysql --enable-large-files-without-
debug
#make
#make install
```

MySQL is installed, and then we need to create a group for MySQL as follows:

```
#!/usr/sbin/groupadd mysql
```

And create a user called mysql which belongs to the mysql group:

```
#!/usr/sbin/useradd -g mysql mysql
```

Now we install the database files as follows:

```
#./scripts/mysql_install_db
```

Then we make a couple minor ownership changes:

```
# chown -R root:mysql /usr/local/mysql
```

```
# chown -R mysql:mysql /usr/local/mysql/data
```

Last but not least, we use add a line the ld.so.conf file as follows;

```
#vi /etc/ld.so.conf
```

And we add the following line;

```
/usr/local/mysql/lib/mysql
```

Now MySQL is installed, we can run it by issuing the following command:

```
#!/usr/local/mysql/bin/mysqld_safe -user=mysql &
```

And as long as we're here we might as well set a root password for MySQL as follows:

```
#!/usr/local/mysql/bin/mysqladmin -u root password  
new_password
```

Where "new\_password" is the password we want to use.

#### 5.1.4 Install Personal Home Page

Like installing MySQL step, you also need a 'root' right to install Personal Home Page (PHP). Go into the PHP source directory as follows:

```
#cd php-4.3.4
```

Follow these commands by typing:

```
#./configure --prefix=/usr/local/php --with-  
apxs2=/usr/local/apache2/bin/apxs --with-  
mysql=/usr/local/mysql  
#make  
#make install  
#cp php.ini-recommended /usr/local/php/lib/php.ini
```

Then edit that file;

```
#vi /usr/local/php/lib/php.ini
```

And change the following:

Find the "doc\_root" section and enter the correct path  
for the directory which serves your web content, such as:

```
doc_root= "/usr/local/apache2/htdocs/"
```

(this is default for apache2)

Then find the file\_uploads section and change it to  
reflect the following:

```
file_uploads=Off
```

So we are done for PHP.

#### 5.1.5 Install Open Database Connectivity

iODBC is a library of functions that implements the  
Open Database Connectivity (ODBC) protocol. It is mainly  
used to database engines running on Microsoft Windows. In  
the Linux operating system, the installation ODBC is

optional, so we skip it.

## 5.2 Backup and Restore

Backup is very important for maintaining the system. Once the operating system fails, the backup files will play a main role for recovering the system.

### 5.2.1 System Backup

Assume all CROS system files are located in the directory"/www". We can type the following command to do system backup.

```
tar -cf www.tar /www
```

In the future, we can use the www.tar to restore the CROS system.

### 5.2.2 Database Backup

To backup the database system, we can use the following command to backup the database:

```
mysqldump CROS | gzip > crosdb.zip
```

After done the command above, the file crosdb.zip would be the backup file of the database.

### 5.2.3 System Restore

To restore the CROS system file, we can type the following command:

```
#tar -xvf www.tar /
```

Now all CROS system files will restore into the



directory /www.

#### 5.2.4 Database Restore

To restore the database, we can issue the following commands:

```
mysqladmin create CROS
```

```
cat www/*.sql | mysql CROS
```

```
mysqlimport CROS www/*.txt
```

Now the database is restored to the database system.

## CHAPTER SIX

### CONCLUSION AND FUTURE DIRECTIONS

#### 6.1 Conclusion

CROS is a more function Web application for the Chinese restaurant. It can provide an efficient communication environment for the consumer and Chinese restaurant over the Internet. With it, the consumer can order some dishes and view some information from the Chinese restaurant Web site. For the restaurant, the employee can efficiently use the CROS to run the business. For instance, the Chinese restaurant staff can use it to process some services and manage the business. Just about service consumer order from a traditional storefront also can be ordered on the Internet. Meanwhile it is also a useful management tool for Chinese restaurant. With the CROS, the operator can efficiently process the orders. The manager can use it to manage the business since it can automatically generate some period sale and analysis reports. With the CROS, transactions and management can occur instantaneously and globally, thereby saving time for both the consumer and business.

## 6.2 Future Directions

The CROS is designed to implement a stand-alone Chinese restaurant to run its business online. However, for the other restaurants rather than the Chinese ones, their business rules and regulations are very similar with Chinese ones. Therefore, by slightly modifying the CROS, it will be able to be reused in various restaurants rather than limiting Chinese restaurant business.

In addition, there are a lot of chain fast food restaurants across the state, how to modify the stand-alone CROS to implement a chain restaurant online system will should be a main consideration issue in the future.

APPENDIX A  
CHINESE RESTAURANT ONLINE SYSTEM  
FILES PRINTOUT

```

<?php

/*****
Filename: AdminDishes.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");
include ("./function.php");

session_start();

$filename = "AdminDishes.php";

check_security(2);

$tpl = new Template($app_path);
$tpl->load_file("AdminDishes.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Search_show();
Items_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "AdminDishes.php");

    // Set variables with search parameters
    $fldcategory_id = strip(get_param("category_id"));
    $fldis_recommended = strip(get_param("is_recommended"));
    // Show fields
    LB_non2(LBcategory_id, $dbccategory_id, "category_id",
    $fldcategory_id, "categories", "name");

    $tpl->set_var("LBis_recommended", "");
    $LOV = split(";", "All;0;No;1;Yes");

    if (sizeof($LOV)%2 != 0)
        $array_length = sizeof($LOV) - 1;
    else
        $array_length = sizeof($LOV);
    reset($LOV);
    for ($i = 0; $i < $array_length; $i = $i + 2)
    {

```

```

    $tpl->set_var("ID", $LOV[$i]);
    $tpl->set_var("Value", $LOV[$i + 1]);
    if ($LOV[$i] == $fldis_recommended)
        $tpl->set_var("Selected", "SELECTED");
    else
        $tpl->set_var("Selected", "");
    $tpl->parse("LBis_recommended", true);
}
$tpl->parse("FormSearch", false);
}

function Items_show()
{
    global $tpl;
    global $db;
    global $sItemsErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "category_id=" .
tourl(strip(get_param("category_id"))) .
    "&is_recommended=" . tourl(strip(get_param("is_recommended"))) .
    "&");
    $tpl->set_var("FormParams", "category_id=" .
tourl(strip(get_param("category_id"))) .
    "&is_recommended=" . tourl(strip(get_param("is_recommended"))) .
    "&");
    // Build WHERE statement

    $pcategory_id = get_param("category_id");
    if (is_number($pcategory_id) && strlen($pcategory_id))
        $pcategory_id = round($pcategory_id);
    else
        $pcategory_id = "";
    if (strlen($pcategory_id))
    {
        $HasParam = true;
        $sWhere .= "i.category_id=" . $pcategory_id;
    }
    $pis_recommended = get_param("is_recommended");
    if (is_number($pis_recommended) && strlen($pis_recommended))
        $pis_recommended = round($pis_recommended);
    else
        $pis_recommended = "";
    if (strlen($pis_recommended))
    {
        if ($sWhere != "") $sWhere .= " and ";
        $HasParam = true;
        $sWhere .= "i.is_recommended=" . $pis_recommended;
    }
    if ($HasParam)
        $sWhere = " AND (" . $sWhere . ")";

    $sDirection = "";
    $sSortParams = "";

```

```

// Build ORDER statement
$iSort = get_param("FormItems_Sorting");
$iSorted = get_param("FormItems_Sorted");
if (!$iSort)
    $tpl->set_var("Form_Sorting", "");
else
{
    if ($iSort == $iSorted)
    {
        $tpl->set_var("Form_Sorting", "");
        $sDirection = " DESC";
        $sSortParams = "FormItems_Sorting=" . $iSort .
"&FormItems_Sorted=" . $iSort . "&";
    }
    else
    {
        $tpl->set_var("Form_Sorting", $iSort);
        $sDirection = " ASC";
        $sSortParams = "FormItems_Sorting=" . $iSort .
"&FormItems_Sorted=" . "&";
    }

    if ($iSort == 1) $sOrder = " order by i.name" . $sDirection;
    if ($iSort == 2) $sOrder = " order by i.price" . $sDirection;
    if ($iSort == 3) $sOrder = " order by c.name" . $sDirection;
    if ($iSort == 4) $sOrder = " order by i.is_recommended" .
$sDirection;
}

// Build full SQL statement
$sSQL = "select i.category_id as i_category_id, " .
    "i.is_recommended as i_is_recommended, " .
    "i.item_id as i_item_id, " .
    "i.name as i_name, i.price as i_price, " .
    "c.category_id as c_category_id, " .
    "c.name as c_name from items i, categories c" .
    " where c.category_id=i.category_id ";
$sSQL .= $sWhere . $sOrder;
$tpl->set_var("FormAction", "DishMaint.php");
$tpl->set_var("SortParams", $sSortParams);

// Execute SQL statement
$db->query($sSQL);

// Select current page
$iPage = get_param("FormItems_Page");
if (!strlen($iPage)) $iPage = 1;
$RecordsPerPage = 10;
if (($iPage - 1) * $RecordsPerPage != 0)
    $db->seek(($iPage - 1) * $RecordsPerPage);
$iCounter = 0;
$ais_recommended = split(";", "0;No;1;Yes");
if ($db->next_record())
{
    // Show main table based on SQL query
    do {

```

```

        $fldname = $db->f("i_name");
        $fldprice = number_format($db->f("i_price"), 2);
        $fldcategory_id = $db->f("c_name");
        $fldis_recommended = $db->f("i_is_recommended");
        $fldField1= "Edit";
        $tpl->set_var("Field1", tohtml($fldField1));
        $tpl->set_var("Field1_URLLink", "DishMaint.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->set_var("name", tohtml($fldname));
        $tpl->set_var("price", tohtml($fldprice));
        $tpl->set_var("category_id", tohtml($fldcategory_id));
        $fldis_recommended = get_lov_value($fldis_recommended,
$ais_recommended);
        $tpl->set_var("is_recommended", tohtml($fldis_recommended));
        $tpl->parse("DListItems", true);
        $iCounter++;
    } while ($iCounter < $RecordsPerPage &&$db->next_record());
}
else
{
    $tpl->set_var("DListItems", "");
    $tpl->parse("ItemsNoRecords", false);
    $tpl->set_var("ItemsScroller", "");
    $tpl->parse("FormItems", false);
    return;
}
if (@$db->next_record())
{
    if ($iPage == 1)
        $tpl->set_var("ItemsScrollerPrevSwitch", "_");
    else
    {
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("ItemsScrollerPrevSwitch", "");
    }
    $tpl->set_var("NextPage", ($iPage + 1));
    $tpl->set_var("ItemsScrollerNextSwitch", "");
    $tpl->set_var("ItemsCurrentPage", $iPage);
    $tpl->parse("ItemsScroller", false);
}
else
{
    if ($iPage == 1)
        $tpl->set_var("ItemsScroller", "");
    else
    {
        $tpl->set_var("ItemsScrollerNextSwitch", "_");
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("ItemsScrollerPrevSwitch", "");
        $tpl->set_var("ItemsCurrentPage", $iPage);
        $tpl->parse("ItemsScroller", false);
    }
}
$tpl->set_var("ItemsNoRecords", "");
$tpl->parse("FormItems", false);
}
?>

```



```

<?php
/*****
 *      Filename: AdvSearch.php
 *****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");
include ("./function.php");

session_start();

$filename = "AdvSearch.php";

$tpl = new Template($app_path);
$tpl->load_file("AdvSearch.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Search_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "SubMenu.php");

    // Set variables with search parameters
    $fldname = strip(get_param("name"));
    $fldcategory_id = strip(get_param("category_id"));
    $fldpricemin = strip(get_param("pricemin"));
    $fldpricemax = strip(get_param("pricemax"));
    // Show fields
    $tpl->set_var("name", tohtml($fldname));
    LB(LBcategory_id, $dbccategory_id, "category_id", $fldcategory_id,
"categories", "name");
    $tpl->set_var("pricemin", tohtml($fldpricemin));
    $tpl->set_var("pricemax", tohtml($fldpricemax));
    $tpl->parse("FormSearch", false);
}
?>

```

```

<?php
/*****
*      Filename: AnalysisGrid.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");
include ("./manager.php");

session_start();

$filename = "AnalysisGrid.php";

check_security(2);

$tpl = new Template($app_path);
$tpl->load_file("AnalysisGrid.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Search_show();
Reports_show();
Report_Show("FormItems1", "AnalysisItemGrid.php");
Report_Show("FormItems2", "AnalysisOrderGrid.php");
Report_Show("FormItems3", "AnalysisGrid.php");
Report_Show("FormItems4", "AnalysisMonthSaleGrid.php");
Report_Show("FormItems5", "AnalysisYearSaleGrid.php");
Report_Show("FormItems6", "AnalysisCustomerGrid.php");

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "AnalysisGrid.php");
    // Set variables with search parameters
    $flddate = strip(get_param("date"));
    // Show fields
    $tpl->set_var("date", tohtml($flddate));
    $tpl->parse("FormSearch", false);
}

function Reports_show()
{
    global $tpl;
    global $db;
    global $sItemsErr;
    $sWhere = "";

```

```

$sOrder = "";
$sSQL = "";
$HasParam = false;

$tpl->set_var("TransitParams", "date1=" .
tourl(strip(get_param("date"))) . "&");
$tpl->set_var("FormParams", "date1=" .
tourl(strip(get_param("date"))) . "&");
// Build WHERE statement

$update = get_param("date");
if (strlen($update))
{
    $HasParam = true;
    $sWhere .= "r.date1=" . tosql($update, "Text");
}

if ($HasParam)
    $sWhere = " where (" . $sWhere . ")";

$sDirection = "";
$sSortParams = "";

// Build ORDER statement
$sOrder = " order by r.report_id DESC";
$iSort = get_param("FormItems_Sorting");
$iSorted = get_param("FormItems_Sorted");
if (!$iSort)
    $tpl->set_var("Form_Sorting", "");
else
{
    if ($iSort == $iSorted)
    {
        $tpl->set_var("Form_Sorting", "");
        $sDirection = " DESC";
        $sSortParams = "FormItems_Sorting=" . $iSort .
"&FormItems_Sorted=" . $iSort . "&";
    }
    else
    {
        $tpl->set_var("Form_Sorting", $iSort);
        $sDirection = " ASC";
        $sSortParams = "FormItems_Sorting=" . $iSort .
"&FormItems_Sorted=" . "&";
    }
    if ($iSort == 1) $sOrder = " order by date1" . $sDirection;
    if ($iSort == 2) $sOrder = " order by field1" . $sDirection;
    if ($iSort == 3) $sOrder = " order by r.field2" . $sDirection;
    if ($iSort == 4) $sOrder = " order by r.field3" . $sDirection;
    if ($iSort == 5) $sOrder = " order by r.field4" . $sDirection;
    if ($iSort == 6) $sOrder = " order by r.field5" . $sDirection;
    if ($iSort == 7) $sOrder = " order by r.field6" . $sDirection;
    if ($iSort == 8) $sOrder = " order by r.field7" . $sDirection;
    if ($iSort == 9) $sOrder = " order by r.field8" . $sDirection;
    if ($iSort == 10) $sOrder = " order by r.field9" . $sDirection;
    if ($iSort == 12) $sOrder = " order by r.field11" . $sDirection;
    if ($iSort == 13) $sOrder = " order by r.field12" . $sDirection;
}

```

```

    if ($iSort == 14) $sOrder = " order by r.field13" . $sDirection;
    if ($iSort == 15) $sOrder = " order by r.field14" . $sDirection;
    if ($iSort == 16) $sOrder = " order by r.field15" . $sDirection;
    if ($iSort == 17) $sOrder = " order by r.field16" . $sDirection;
}

// Build full SQL statement
$sSQL = "select * from reports r";

$sSQL .= $sWhere . $sOrder;
$tpl->set_var("FormAction", "ReportRecord.php");
$tpl->set_var("SortParams", $sSortParams);

// Execute SQL statement
$db->query($sSQL);

// Select current page
$iPage = get_param("FormItems_Page");
if (!strlen($iPage)) $iPage = 1;
$RecordsPerPage = 10;
if (($iPage - 1) * $RecordsPerPage != 0)
    $db->seek(($iPage - 1) * $RecordsPerPage);
$iCounter = 0;
$i = 1;
$fldmaxdate = "2004-01-01";
if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldfield1 = $db->f("date1");
        if (!strlen($fldmindate))
        {
            $fldmindate = $fldfield1;
            $fldmaxdate = $fldfield1;
        }
        if ($fldmaxdate < $fldfield1)
            $fldmaxdate = $fldfield1;
        if ($fldmindate > $fldfield1)
            $fldmindate = $fldfield1;
        $fldfield2 = $db->f("field1");
        $fldfield201 += $fldfield2;
        $fldfield3 = $db->f("field2");
        $fldfield301 += $fldfield3;
        $fldfield4 = $db->f("field3");
        $fldfield401 += $fldfield4;
        $fldfield5 = $db->f("field4");
        $fldfield501 += $fldfield5;
        $fldfield6 = $db->f("field5");
        $fldfield601 += $fldfield6;
        $fldfield7 = $db->f("field6");
        $fldfield701 += $fldfield7;
        $fldfield8 = $db->f("field7");
        $fldfield801 += $fldfield8;
        $fldfield9 = number_format($db->f("field8"), 2);
        $fldfield901 += $fldfield9;
        $fldfield10 = $db->f("field9");
    }
}

```

```

$fldfield1101 += $fldfield10;
$fldfield12 = $db->f("field11");
$fldfield1201 += $fldfield12;
$fldfield13 = $db->f("field12");
$fldfield1301 += $fldfield13;
$fldfield14 = $db->f("field13");
$fldfield1401 += $fldfield14;
$fldfield15 = $db->f("field14");
$fldfield1501 += $fldfield15;
$fldfield16 = $db->f("field15");
$fldfield1601 += $fldfield16;
$fldfield17 = $db->f("field16");
$fldfield1701 += $fldfield17;
if (strlen($sWhere))
{
    $i = ""; $orde = "";
}
else
{
    $i=($iPage-1)*10+$iCounter+1;
    $orde = "Order";
}
$tpl->set_var("orde", tohtml($orde));
$tpl->set_var("order_id", tohtml($i));
$fldField1="Print";
$tpl->set_var("Field1", tohtml($fldField1));
$tpl->set_var("Field1_URLLink", "PrintSaleReport.php");
$tpl->set_var("Prm_item_id", tourl($db->f("date1")));
$tpl->set_var("field1", tohtml($fldfield1));
$tpl->set_var("field2", tohtml($fldfield2));
$tpl->set_var("field3", tohtml($fldfield3));
$tpl->set_var("field4", tohtml($fldfield4));
$tpl->set_var("field5", tohtml($fldfield5));
$tpl->set_var("field6", tohtml($fldfield6));
$tpl->set_var("field7", tohtml($fldfield7));
$tpl->set_var("field8", tohtml($fldfield8));
$tpl->set_var("field9", tohtml($fldfield9));
$tpl->set_var("field10", tohtml($fldfield10));
$tpl->set_var("field12", tohtml($fldfield12));
$tpl->set_var("field13", tohtml($fldfield13));
$tpl->set_var("field14", tohtml($fldfield14));
$tpl->set_var("field15", tohtml($fldfield15));
$tpl->set_var("field16", tohtml($fldfield16));
$tpl->set_var("field17", tohtml($fldfield17));
$tpl->set_var("field201", tohtml($fldfield201));
$tpl->set_var("field301", tohtml($fldfield301));
$tpl->set_var("field401", tohtml($fldfield401));
$tpl->set_var("field501", tohtml($fldfield501));
$tpl->set_var("field601", tohtml($fldfield601));
$tpl->set_var("field701", tohtml($fldfield701));
$tpl->set_var("field801", tohtml($fldfield801));
$tpl->set_var("field901", tohtml($fldfield901));
$tpl->set_var("field1101", tohtml($fldfield1101));
$tpl->set_var("field1201", tohtml($fldfield1201));
$tpl->set_var("field1301", tohtml($fldfield1301));
$tpl->set_var("field1401", tohtml($fldfield1401));
$tpl->set_var("field1501", tohtml($fldfield1501));

```

```

        $tpl->set_var("field1601", tohtml($fldfield1601));
        $tpl->set_var("field1701", tohtml($fldfield1701));
        $tpl->set_var("name_URLLink", "AnalysisMonthSaleRecord.php");
        $tpl->parse("DListItems", true);
        $iCounter++;
    } while ($iCounter < $RecordsPerPage &&$db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListItems", "");
    $tpl->parse("ItemsNoRecords", false);
    $tpl->set_var("ItemsScroller", "");
    $tpl->parse("FormItems", false);
    return;
}

if (@$db->next_record())
{
    if ($iPage == 1)
        $tpl->set_var("ItemsScrollerPrevSwitch", "_");
    else
    {
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("ItemsScrollerPrevSwitch", "");
    }
    $tpl->set_var("NextPage", ($iPage + 1));
    $tpl->set_var("ItemsScrollerNextSwitch", "");
    $tpl->set_var("ItemsCurrentPage", $iPage);
    $tpl->parse("ItemsScroller", false);
}
else
{
    if ($iPage == 1)
        $tpl->set_var("ItemsScroller", "");
    else
    {
        $tpl->set_var("ItemsScrollerNextSwitch", "_");
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("ItemsScrollerPrevSwitch", "");
        $tpl->set_var("ItemsCurrentPage", $iPage);
        $tpl->parse("ItemsScroller", false);
    }
}

$fldmindatel = substr($fldmindate, 5,2) . "/" . substr($fldmindate,
8,2) . "/" . substr($fldmindate, 2,2);
$fldmaxdatel = substr($fldmaxdate, 5,2) . "/" . substr($fldmaxdate,
8,2) . "/" . substr($fldmaxdate, 2,2);
$fldfield101 = $fldmindatel . "-" . $fldmaxdatel;
$tpl->set_var("field101", tohtml($fldfield101));
$fldmin = dlookup("reports", "min(date_format(datel, '%m/%d/%y'))",
"report_id>0");
$fldmax = dlookup("reports", "max(date_format(datel, '%m/%d/%y'))",
"report_id>0");
$fldfield2011 = dlookup("reports", "sum(field1)", "report_id>0");
$fldfield3011 = dlookup("reports", "sum(field2)", "report_id>0");
$fldfield4011 = dlookup("reports", "sum(field3)", "report_id>0");
$fldfield5011 = dlookup("reports", "sum(field4)", "report_id>0");
$fldfield6011 = dlookup("reports", "sum(field5)", "report_id>0");

```

```

$fldfield7011 = dlookup("reports", "sum(field6)", "report_id>0");
$fldfield8011 = dlookup("reports", "sum(field7)", "report_id>0");
$fldfield9011 = dlookup("reports", "sum(field8)", "report_id>0");
$fldfield11011 = dlookup("reports", "sum(field9)", "report_id>0");
$fldfield12011 = dlookup("reports", "sum(field11)", "report_id>0");
$fldfield13011 = dlookup("reports", "sum(field12)", "report_id>0");
$fldfield14011 = dlookup("reports", "sum(field13)", "report_id>0");
$fldfield15011 = dlookup("reports", "sum(field14)", "report_id>0");
$fldfield16011 = dlookup("reports", "sum(field15)", "report_id>0");
$fldfield17011 = dlookup("reports", "sum(field16)", "report_id>0");
$fldtotalday = dlookup("reports", "count(*)", "report_id>0") . "
days";
$tpl->set_var("ItemsNoRecords", "");
$fldfield1011 = $fldmin . "-" . $fldmax;
$tpl->set_var("totalday", tohtml($fldtotalday));
$tpl->set_var("field1011", tohtml($fldfield1011));
$tpl->set_var("field2011", tohtml($fldfield2011));
$tpl->set_var("field3011", tohtml($fldfield3011));
$tpl->set_var("field4011", tohtml($fldfield4011));
$tpl->set_var("field5011", tohtml($fldfield5011));
$tpl->set_var("field6011", tohtml($fldfield6011));
$tpl->set_var("field7011", tohtml($fldfield7011));
$tpl->set_var("field8011", tohtml($fldfield8011));
$tpl->set_var("field9011", tohtml($fldfield9011));
$tpl->set_var("field11011", tohtml($fldfield11011));
$tpl->set_var("field12011", tohtml($fldfield12011));
$tpl->set_var("field13011", tohtml($fldfield13011));
$tpl->set_var("field14011", tohtml($fldfield14011));
$tpl->set_var("field15011", tohtml($fldfield15011));
$tpl->set_var("field16011", tohtml($fldfield16011));
$tpl->set_var("field17011", tohtml($fldfield17011));
$tpl->parse("FormItems", false);
}
?>

```

```

<?php
/*****
*      Filename: CheckOrders.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");
include ("./manager.php");

session_start();

$filename = "CheckOrders.php";

check_security(2);

$tpl = new Template($app_path);
$tpl->load_file("CheckOrders.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Search_show();
Orders_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "CheckOrders.php");

    // Set variables with search parameters
    $fldname = strip(get_param("name"));
    // Show fields
    $tpl->set_var("name", tohtml($fldname));
    $tpl->parse("FormSearch", false);
}

function Orders_show()
{
    global $tpl;
    global $db;
    global $sMembersErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "name=" .

```



```

tourl(strip(get_param("name"))) . "&");
$tpl->set_var("FormParams", "name=" .
tourl(strip(get_param("name"))) . "&");
// Build WHERE statement

$pname = get_param("name");
if(strlen($pname))
{
    $HasParam = true;
    $sWhere = "(m.last_name like " . tosql("%".$pname . "%", "Text") .
" or " .
                "m.first_name like " . tosql("%".$pname . "%", "Text")
. ") or (" .
                "phone=" . tosql($pname, "Text") . ") or (" .
                "order_id=" . tosql($pname, "Text") . ")";
}

if($HasParam)
    $sWhere = " AND (" . $sWhere . ")";

$sDirection = "";
$sSortParams = "";

// Build ORDER statement
$sOrder = " order by o.order_id DESC";
$iSort = get_param("FormMembers_Sorting");
$iSorted = get_param("FormMembers_Sorted");
if(!$iSort)
    $tpl->set_var("Form_Sorting", "");
else
{
    if($iSort == $iSorted)
    {
        $tpl->set_var("Form_Sorting", "");
        $sDirection = " DESC";
        $sSortParams = "FormMembers_Sorting=" . $iSort .
"&FormMembers_Sorted=" . $iSort . "&";
    }
    else
    {
        $tpl->set_var("Form_Sorting", $iSort);
        $sDirection = " ASC";
        $sSortParams = "FormMembers_Sorting=" . $iSort .
"&FormMembers_Sorted=" . "&";
    }

    if ($iSort == 1) $sOrder = " order by order_id" . $sDirection;
    if ($iSort == 2) $sOrder = " order by phone" . $sDirection;
    if ($iSort == 3) $sOrder = " order by last_name" . $sDirection;
    if ($iSort == 4) $sOrder = " order by cd_dates" . $sDirection;
    if ($iSort == 5) $sOrder = " order by statu" . $sDirection;
}

// Build full SQL statement
$current = date("Y-m-d ") . "02:00:00";
$sql = "select m.first_name, " .
        "m.last_name, " .

```

```

        "m.member_id as m_member_id, " .
        "m.phone, " .
        "o.order_id as o_order_id, " .
        "o.statu, " .
        "o.cd_dates " .
        "from members m, orders o where o.member_id=m.member_id and
cd_dates>=" .
        tosql($current, "Text") . " ";
    $sSQL .= $sWhere . $sOrder;
    $tpl->set_var("FormAction", "CustomersRecord.php");
    $tpl->set_var("SortParams", $sSortParams);
    // Execute SQL statement
    $db->query($sSQL);
    // Select current page
    $iPage = get_param("FormMembers_Page");
    if(!strlen($iPage)) $iPage = 1;
    $RecordsPerPage = 10;
    if(($iPage - 1) * $RecordsPerPage != 0)
        $db->seek(($iPage - 1) * $RecordsPerPage);
    $iCounter = 0;
    $i = 1;
    if($db->next_record())
    {
        // Show main table based on SQL query
        do
        {
            $fldorder_id = $db->f("o_order_id");
            $fldphone = $db->f("phone");
            $fldtimes = substr($db->f("cd_dates"), 11, 8);
            $fldstatus = $db->f("statu");
            $fldname = $db->f("first_name");
            $fldname .= " " . $db->f("last_name");
            $i = ($iPage-1)*10 + $iCounter+1;
            $tpl->set_var("n", tohtml($i));
            $tpl->set_var("order_id", tohtml($fldorder_id));
            $tpl->set_var("member_login_URLLink", "CustomersInfo.php");
            $tpl->set_var("invoice_URLLink", "Invoice.php");
            $tpl->set_var("Prm_order_id", tourl($db->f("o_order_id")));
            $tpl->set_var("phone", tohtml($fldphone));
            if ($fldstatus==1)
            {
                $fldstatus = "Prepare Order";
                $fldprocess = "";
            }
            if ($fldstatus==2)
            {
                $fldstatus = "Ordered";
                $fldprocess = "Print Order";
            }
            if ($fldstatus==3)
            {
                $fldstatus = "Paid";
                $fldprocess = "Print Invoice";
            }
            if ($fldstatus==4)
                $fldstatus .= " Printed";
            if ($fldstatus==5)

```

```

        $fldstatus .= " Modified";
        if ($fldstatus==6)
            $fldstatus .= " Done";
        $tpl->set_var("status", tohtml($fldstatus));
        $tpl->set_var("process", tohtml($fldprocess));
        $tpl->set_var("last_name", tohtml($fldname));
        $tpl->set_var("times", tohtml($fldtimes));
        $tpl->parse("DListMembers", true);
        $iCounter++;
    } while($iCounter < $RecordsPerPage &&$db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListMembers", "");
    $tpl->parse("MembersNoRecords", false);
    $tpl->set_var("MembersScroller", "");
    $tpl->parse("FormMembers", false);
    return;
}

// Parse scroller
if(@$db->next_record())
{
    if ($iPage == 1)
        $tpl->set_var("MembersScrollerPrevSwitch", "_");
    else
    {
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("MembersScrollerPrevSwitch", "");
    }
    $tpl->set_var("NextPage", ($iPage + 1));
    $tpl->set_var("MembersScrollerNextSwitch", "");
    $tpl->set_var("MembersCurrentPage", $iPage);
    $tpl->parse("MembersScroller", false);
}
else
{
    if ($iPage == 1)
        $tpl->set_var("MembersScroller", "");
    else
    {
        $tpl->set_var("MembersScrollerNextSwitch", "_");
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("MembersScrollerPrevSwitch", "");
        $tpl->set_var("MembersCurrentPage", $iPage);
        $tpl->parse("MembersScroller", false);
    }
}
$tpl->set_var("MembersNoRecords", "");
$tpl->parse("FormMembers", false);
}
?>

```

```

<?php
/*****
*      Filename: CustomersGrid.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");

session_start();

$filename = "CustomersGrid.php";

check_security(2);

$tpl = new Template($app_path);
$tpl->load_file("CustomersGrid.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Search_show();
Customers_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "CustomersGrid.php");

    // Set variables with search parameters
    $fldname = strip(get_param("name"));
    // Show fields
    $tpl->set_var("name", tohtml($fldname));
    $tpl->parse("FormSearch", false);
}

function Customers_show()
{
    global $tpl;
    global $db;
    global $sCustomersErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "name=" .
    tourl(strip(get_param("name"))) . "&");

```

```

$tpl->set_var("FormParams", "name=" .
tourl(strip(get_param("name")) . "&");
// Build WHERE statement

$name = get_param("name");
if (strlen($name))
{
    $HasParam = true;
    $sWhere = "o.order_id=" . tosql($name, "Text");
}

if ($HasParam)
    $sWhere = " WHERE (" . $sWhere . ")";

$sDirection = "";
$sSortParams = "";

// Build ORDER statement
$sOrder = " order by order_id DESC";
$iSort = get_param("FormCustomers_Sorting");
$iSorted = get_param("FormCustomers_Sorted");
if (!$iSort)
    $tpl->set_var("Form_Sorting", "");
else
{
    if ($iSort == $iSorted)
    {
        $tpl->set_var("Form_Sorting", "");
        $sDirection = " DESC";
        $sSortParams = "FormCustomers_Sorting=" . $iSort .
"&FormCustomers_Sorted=" . $iSort . "&";
    }
    else
    {
        $tpl->set_var("Form_Sorting", $iSort);
        $sDirection = " ASC";
        $sSortParams = "FormCustomers_Sorting=" . $iSort .
"&FormCustomers_Sorted=" . "&";
    }
    if ($iSort == 1) $sOrder = " order by member_id" . $sDirection;
    if ($iSort == 2) $sOrder = " order by statu" . $sDirection;
    if ($iSort == 3) $sOrder = " order by servern" . $sDirection;
    if ($iSort == 4) $sOrder = " order by guestn" . $sDirection;
    if ($iSort == 5) $sOrder = " order by tablen" . $sDirection;
    if ($iSort == 6) $sOrder = " order by cd_dates" . $sDirection;
}

// Build full SQL statement
$sSQL = "select * from orders o ";
$sSQL .= $sWhere . $sOrder;
$tpl->set_var("FormAction", "CustomersRecord.php");
$tpl->set_var("SortParams", $sSortParams);

// Execute SQL statement
$db->query($sSQL);

// Select current page

```

```

$iPage = get_param("FormCustomers_Page");
if (!strlen($iPage)) $iPage = 1;
$RecordsPerPage = 10;
if (($iPage - 1) * $RecordsPerPage != 0)
    $db->seek(($iPage - 1) * $RecordsPerPage);
$iCounter = 0;

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldorder_id = $db->f("order_id");
        $fldservern = $db->f("servern");
        $fldtablen = $db->f("tablen");
        $fldguestn = $db->f("guestn");
        $fldstatu = $db->f("statu");
        if ($fldstatu == 0)
            $fldstatu = "Prepare Order";
        elseif ($fldstatu == 1)
            $fldstatu = "Confirm Order";
        elseif ($fldstatu == 2)
            $fldstatu = "Ordered";
        elseif ($fldstatu == 3)
            $fldstatu = "Paid";
        $fldtimes = $db->f("cd_dates");
        $tpl->set_var("member_id", tohtml($fldorder_id));
        $tpl->set_var("member_id_URLLink", "CustomersInfo.php");
        $tpl->set_var("Prm_member_id", tourl($db->f("order_id")));
        $tpl->set_var("servern", tohtml($fldservern));
        $tpl->set_var("guestn", tohtml($fldguestn));
        $tpl->set_var("statu", tohtml($fldstatu));
        $tpl->set_var("tablen", tohtml($fldtablen));
        $tpl->set_var("times", tohtml($fldtimes));
        $tpl->parse("DListCustomers", true);
        $iCounter++;
    } while ($iCounter < $RecordsPerPage &&$db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListCustomers", "");
    $tpl->parse("CustomersNoRecords", false);
    $tpl->set_var("CustomersScroller", "");
    $tpl->parse("FormCustomers", false);
    return;
}

// Parse scroller
if (@$db->next_record())
{
    if ($iPage == 1)
        $tpl->set_var("CustomersScrollerPrevSwitch", "_");
    else
    {
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("CustomersScrollerPrevSwitch", "");
    }
}

```

```

    }
    $tpl->set_var("NextPage", ($iPage + 1));
    $tpl->set_var("CustomersScrollerNextSwitch", "");
    $tpl->set_var("CustomersCurrentPage", $iPage);
    $tpl->parse("CustomersScroller", false);
}
else
{
    if ($iPage == 1)
        $tpl->set_var("CustomersScroller", "");
    else
    {
        $tpl->set_var("CustomersScrollerNextSwitch", "_");
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("CustomersScrollerPrevSwitch", "");
        $tpl->set_var("CustomersCurrentPage", $iPage);
        $tpl->parse("CustomersScroller", false);
    }
}
$tpl->set_var("CustomersNoRecords", "");
$tpl->parse("FormCustomers", false);
}
?>

```

```

<?php
/*****
*      Filename: Default.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");
include ("./function.php");

session_start();

$filename = "Default.php";

$tpl = new Template($app_path);
$tpl->load_file("Default.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Search_show();
AdvMenu_show();
Recommended_show();
What_show();
Categories_show();
New_show();
Weekly_show();
Specials_show();
Recommended1_show();
Recommended2_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "SubMenu.php");

    // Set variables with search parameters
    $fldcategory_id = strip(get_param("category_id"));
    $fldname = strip(get_param("name"));
    // Show fields
    LB_non2(LBcategory_id, $dbccategory_id, "category_id",
    $fldcategory_id, "categories", "name");

    $tpl->set_var("name", tohtml($fldname));
    $tpl->parse("FormSearch", false);
}

function AdvMenu_show()

```



```

{
    global $tpl;
    // Set URLs
    $fldField1 = "AdvSearch.php";
    // Show fields
    $tpl->set_var("Field1", $fldField1);
    $tpl->parse("FormAdvMenu", false);
}

function Recommended_show()
{
    global $tpl;
    global $db;
    global $sRecommendedErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    // Build WHERE statement

    $sWhere = " WHERE is_recommended=1";

    $sDirection = "";
    $sSortParams = "";

    // Build ORDER statement
    $iSort = get_param("FormRecommended_Sorting");
    $iSorted = get_param("FormRecommended_Sorted");
    if (!$iSort)
        $tpl->set_var("Form_Sorting", "");
    else
    {
        if ($iSort == $iSorted)
        {
            $tpl->set_var("Form_Sorting", "");
            $sDirection = " DESC";
            $sSortParams = "FormRecommended_Sorting=" . $iSort .
"&FormRecommended_Sorted=" . $iSort . "&";
        }
        else
        {
            $tpl->set_var("Form_Sorting", $iSort);
            $sDirection = " ASC";
            $sSortParams = "FormRecommended_Sorting=" . $iSort .
"&FormRecommended_Sorted=" . "&";
        }

        if ($iSort == 1) $sOrder = " order by i.name" . $sDirection;
        if ($iSort == 2) $sOrder = " order by i.image_url" . $sDirection;
        if ($iSort == 3) $sOrder = " order by i.price" . $sDirection;
    }

    // Build full SQL statement

```

```

$ssSQL = "select i.image_url as i_image_url, " .
        "i.item_id as i_item_id, i.name as i_name, " .
        "i.notes as i_notes from items i ";

$ssSQL .= $sWhere . $sOrder;
$tpl->set_var("SortParams", $sSortParams);

// Execute SQL statement
$db->query($ssSQL);

// Select current page
$iPage = get_param("FormRecommended_Page");
if (!strlen($iPage)) $iPage = 1;
$RecordsPerPage = 5;
if (($iPage - 1) * $RecordsPerPage != 0)
    $db->seek(($iPage - 1) * $RecordsPerPage);
$iCounter = 0;

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldname = $db->f("i_name");
        $fldimage_url = $db->f("i_image_url");
        $fldnotes = $db->f("i_notes");
        $fldimage_url = "";
        $tpl->set_var("name", tohtml($fldname));
        $tpl->set_var("name_URLLink", "DishDetail.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->set_var("image_url", $fldimage_url);
        $tpl->set_var("image_url_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->set_var("notes", tohtml($fldnotes));
        $tpl->parse("DListRecommended", true);
        $iCounter++;
    } while ($iCounter < $RecordsPerPage &&$db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListRecommended", "");
    $tpl->parse("RecommendedNoRecords", false);
    $tpl->set_var("RecommendedScroller", "");
    $tpl->parse("FormRecommended", false);
    return;
}

// Parse scroller
if (@$db->next_record())
{
    if ($iPage == 1)
        $tpl->set_var("RecommendedScrollerPrevSwitch", "_");
    else
    {
        $tpl->set_var("PrevPage", ($iPage - 1));
    }
}

```

```

        $tpl->set_var("RecommendedScrollerPrevSwitch", "");
    }
    $tpl->set_var("NextPage", ($iPage + 1));
    $tpl->set_var("RecommendedScrollerNextSwitch", "");
    $tpl->set_var("RecommendedCurrentPage", $iPage);
    $tpl->parse("RecommendedScroller", false);
}
else
{
    if ($iPage == 1)
        $tpl->set_var("RecommendedScroller", "");
    else
    {
        $tpl->set_var("RecommendedScrollerNextSwitch", "");
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("RecommendedScrollerPrevSwitch", "");
        $tpl->set_var("RecommendedCurrentPage", $iPage);
        $tpl->parse("RecommendedScroller", false);
    }
}
$tpl->set_var("RecommendedNoRecords", "");
$tpl->parse("FormRecommended", false);
}

function What_show()
{
    global $tpl;
    global $db;
    global $sWhatErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    // Build WHERE statement

    $sWhere = " AND editorial_cat_id=1";

    $sDirection = "";
    $sSortParams = "";

    // Build full SQL statement

    $sSQL = "select e.article_desc as e_article_desc, " .
        "e.article_title as e_article_title, e.item_id as e_item_id, " .
        "i.item_id as i_item_id, i.notes as i_notes, " .
        "i.image_url as i_image_url from editorials e, items i" .
        " where i.item_id=e.item_id ";

    $sSQL .= $sWhere . $sOrder;

    // Execute SQL statement
    $db->query($sSQL);

    if ($db->next_record())

```

```

{
    // Show main table based on SQL query
    do
    {
        $fldarticle_title = $db->f("e_article_title");
        $fldarticle_desc = $db->f("e_article_desc");
        $flditem_id = $db->f("i_image_url");
        $fldnotes = $db->f("i_notes");
        $flditem_id="";
        $tpl->set_var("article_title", tohtml($fldarticle_title));
        $tpl->set_var("article_title_URLLink", "DishDetail.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("e_item_id")));
        $tpl->set_var("article_desc", $fldnotes);
        $tpl->set_var("item_id", $flditem_id);
        $tpl->set_var("item_id_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("e_item_id")));
        $tpl->parse("DListWhat", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListWhat", "");
    $tpl->parse("WhatNoRecords", false);
    $tpl->parse("FormWhat", false);
    return;
}

$tpl->set_var("WhatNoRecords", "");
$tpl->parse("FormWhat", false);
}

function Categories_show()
{
    global $tpl;
    global $db;
    global $sCategoriesErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    // Build WHERE statement

    $sDirection = "";
    $sSortParams = "";

    // Build full SQL statement

    $sSQL = "select c.category_id as c_category_id, " .
    "c.name as c_name from categories c ";

    $sSQL .= $sWhere . $sOrder;

```

```

// Execute SQL statement
$db->query($sSQL);

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldname = $db->f("c_name");
        $tpl->set_var("name", tohtml($fldname));
        $tpl->set_var("name_URLLink", "SubMenu.php");
        $tpl->set_var("Prm_category_id", tourl($db-
>f("c_category_id")));
        $tpl->parse("DListCategories", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListCategories", "");
    $tpl->parse("CategoriesNoRecords", false);
    $tpl->parse("FormCategories", false);
    return;
}

$tpl->set_var("CategoriesNoRecords", "");
$tpl->parse("FormCategories", false);
}

function New_show()
{
    global $tpl;
    global $db;
    global $sNewErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    // Build WHERE statement

    $sWhere = " AND editorial_cat_id=2";

    $sDirection = "";
    $sSortParams = "";

    // Build full SQL statement

    $sSQL = "select e.article_desc as e_article_desc, " .
        "e.article_title as e_article_title, " .
        "e.item_id as e_item_id, i.item_id as i_item_id, " .
        "i.notes as i_notes, i.image_url as i_image_url " .
        " from editorials e, items i where i.item_id=e.item_id ";

    $sSQL .= $sWhere . $sOrder;

```

```

// Execute SQL statement
$db->query($sSQL);

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldarticle_title = $db->f("e_article_title");
        $flditem_id = $db->f("i_image_url");
        $fldarticle_desc = $db->f("e_article_desc");
        $fldnotes = $db->f("i_notes");
        $flditem_id = "";
        $tpl->set_var("article_title", tohtml($fldarticle_title));
        $tpl->set_var("article_title_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("e_item_id")));
        $tpl->set_var("item_id", $flditem_id);
        $tpl->set_var("item_id_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("e_item_id")));
        $tpl->set_var("article_desc", tohtml($fldnotes));
        $tpl->parse("DListNew", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListNew", "");
    $tpl->parse("NewNoRecords", false);
    $tpl->parse("FormNew", false);
    return;
}

$tpl->set_var("NewNoRecords", "");
$tpl->parse("FormNew", false);
}

function Weekly_show()
{
    global $tpl;
    global $db;
    global $sWeeklyErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    // Build WHERE statement

    $sWhere = " AND editorial_cat_id=3";

    $sDirection = "";
    $sSortParams = "";

```

```

// Build full SQL statement

$sSQL = "select e.article_desc as e_article_desc, " .
        "e.article_title as e_article_title, " .
        "e.item_id as e_item_id, " .
        "i.item_id as i_item_id, " .
        "i.notes as i_notes, " .
        "i.image_url as i_image_url " .
        " from editorials e, items i" .
        " where i.item_id=e.item_id ";

$sSQL .= $sWhere . $sOrder;

// Execute SQL statement
$db->query($sSQL);

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldarticle_title = $db->f("e_article_title");
        $flditem_id = $db->f("i_image_url");
        $fldarticle_desc = $db->f("e_article_desc");
        $fldnotes = $db->f("i_notes");
        $flditem_id = "";
        $tpl->set_var("article_title", tohtml($fldarticle_title));
        $tpl->set_var("article_title_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("e_item_id")));
        $tpl->set_var("item_id", $flditem_id);
        $tpl->set_var("item_id_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("e_item_id")));
        $tpl->set_var("article_desc", tohtml($fldnotes));
        $tpl->parse("DListWeekly", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListWeekly", "");
    $tpl->parse("WeeklyNoRecords", false);
    $tpl->parse("FormWeekly", false);
    return;
}

$tpl->set_var("WeeklyNoRecords", "");
$tpl->parse("FormWeekly", false);
}

function Specials_show()
{
    global $tpl;
    global $db;
    global $sSpecialsErr;
    $sWhere = "";
    $sOrder = "";

```

```

$ssSQL = "";
$HasParam = false;

$tpl->set_var("TransitParams", "");
$tpl->set_var("FormParams", "");
// Build WHERE statement

$ssWhere = " WHERE editorial_cat_id=4";

$ssDirection = "";
$ssSortParams = "";

// Build full SQL statement

$ssSQL = "select e.article_desc as e_article_desc, " .
        "e.article_title as e_article_title from editorials e ";

$ssSQL .= $ssWhere . $ssOrder;

// Execute SQL statement
$db->query($ssSQL);

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldarticle_title = $db->f("e_article_title");
        $fldarticle_desc = $db->f("e_article_desc");
        $tpl->set_var("article_title", $fldarticle_title);
        $tpl->set_var("article_desc", $fldarticle_desc);
        $tpl->parse("DListSpecials", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListSpecials", "");
    $tpl->parse("SpecialsNoRecords", false);
    $tpl->parse("FormSpecials", false);
    return;
}

$tpl->set_var("SpecialsNoRecords", "");
$tpl->parse("FormSpecials", false);
}

function Recommended1_show()
{
    global $tpl;
    global $db;
    global $sRecommendedErr;
    $ssWhere = "";
    $ssOrder = "";
    $ssSQL = "";
    $HasParam = false;

```



```

$tpl->set_var("TransitParams", "");
$tpl->set_var("FormParams", "");
// Build WHERE statement
$$SQL = "select * from info i where info_id>=2 and info_id<=4 order
by info_id";

// Execute SQL statement
$db->query($$SQL);

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldname = $db->f("name");
        $fldid = $db->f("info_id");
        $fldimage_url = "images/map.jpg";
        $fldimage_url="";
        $tpl->set_var("name", tohtml($fldname));
        $tpl->set_var("name_URLLink", "DishDetail.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->set_var("image_url", $fldimage_url);
        $tpl->set_var("image_url_URLLink", "Map.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->parse("DListRecommended1", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListRecommended1", "");
    $tpl->parse("Recommended1NoRecords", false);
    $tpl->parse("FormRecommended1", false);
    return;
}
$tpl->set_var("Recommended1NoRecords", "");
$tpl->parse("FormRecommended1", false);
}

function Recommended2_show()
{
    global $tpl;
    global $db;
    global $$RecommendedErr;
    $$Where = "";
    $$Order = "";
    $$SQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    // Build WHERE statement
    $$SQL = "select openhour_id, openhour, closehour from openhours";

    // Execute SQL statement
    $db->query($$SQL);

```

```

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldopenhour = $db->f("openhour");
        $fldopenhour = date("h:i A", mktime(substr($fldopenhour,0,2),
substr($fldopenhour,3,2),0,1,1,2004));
        $fldclosehour = $db->f("closehour");
        $fldclosehour = date("h:i A", mktime(substr($fldclosehour,0,2),
substr($fldclosehour,3,2),0,1,1,2004));
        $fldid = $db->f("openhour_id");
        if ($fldid==0)
            $fldname = "SUN: " . $fldopenhour . "--" . $fldclosehour;
        if ($fldid==1)
            $fldname = "MON: " . $fldopenhour . "--" . $fldclosehour;
        if ($fldid==2)
            $fldname = "TUE: " . $fldopenhour . "--" . $fldclosehour;
        if ($fldid==3)
            $fldname = "WED: " . $fldopenhour . "--" . $fldclosehour;
        if ($fldid==4)
            $fldname = "THU: " . $fldopenhour . "--" . $fldclosehour;
        if ($fldid==5)
            $fldname = "FRI: " . $fldopenhour . "--" . $fldclosehour;
        if ($fldid==6)
            $fldname = "SAT: " . $fldopenhour . "--" . $fldclosehour;
        $tpl->set_var("name", tohtml($fldname));
        $tpl->parse("DListRecommended2", true);
    } while ($db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListRecommended2", "");
    $tpl->parse("Recommended2NoRecords", false);
    $tpl->parse("FormRecommended2", false);
    return;
}
$tpl->set_var("Recommended2NoRecords", "");
$tpl->parse("FormRecommended2", false);
}
?>

```

```

<?php
/*****
*      Filename: Header.php
*****/
function Menu_show()
{
    global $tpl;
    // Set URLs
    $fldField2 = "Default.php";
    $fldHome = "Default.php";
    $fldMenu = "Menu.php";
    $fldReg = "Registration.php";
    $fldShop = "ShoppingCart.php";
    $fldField1 = "Login.php";
    // Show fields
    $fldUserRights = get_session("UserRights");
    if ($fldUserRights>1)
    {
        $fldtophr = "<hr width=\"800\" color=\"\#336699\">";
        $fldbottomhr = "";
    }
    else
    {
        $fldtophr = "<hr width=\"640\" color=\"\#FF0000\">";
        $fldbottomhr = "<hr width=\"640\" color=\"\#FF0000\">";
    }
    if ($fldUserRights>4)//administrator
    {
        $fld2 = "OrdersGrid.php";
        $fld3 = "AdminDishes.php";
        $fld4 = "CategoriesGrid.php";
        $fld5 = "EditorialsGrid.php";
        $fld6 = "EditorialCatGrid.php";
        $fld8 = "CustomersGrid.php";
        $fld9 = "CardTypesGrid.php";
        $fld14 = "OpenHourGrid.php";
        $fld15 = "InfoGrid.php";
        $fld102 = "[ Orders ]";
        $fld103 = "[ Dishes ]";
        $fld104 = "[ Categories ]";
        $fld105 = "[ Editorials ]";
        $fld106 = "[ EditorialCat ]";
        $fld108 = "[ Invoices ]";
        $fld109 = "[ Card Types ]";
        $fld1014 = "[ Open Hour ]";
        $fld1015 = "[ Info ]";
    }
    else
    {
        $fld2 = "";
        $fld3 = "";
        $fld4 = "";
        $fld5 = "";
        $fld6 = "";
        $fld8 = "";
        $fld9 = "";
        $fld14 = "";
        $fld15 = "";
    }
}

```

```

    $fld102 = "";
    $fld103 = "";
    $fld104 = "";
    $fld105 = "";
    $fld106 = "";
    $fld108 = "";
    $fld109 = "";
    $fld1014 = "";
    $fld1015 = "";
}
if ($fldUserRights>3)//manager
{
    $fld1 = "MembersGrid.php";
    $fld16 = "AnalysisGrid.php";
    $fld101 = "[ Members ]";
    $fld1016 = "[ Analysis ]";
}
else
{
    $fld1 = "";
    $fld16 = "";
    $fld101 = "";
    $fld1016 = "";
}
if ($fldUserRights>2)//cashier
{
    $fld7 = "CheckOrders.php";
    $fld10 = "Reports.php";
    $fld107 = "[ Check Orders ]";
    $fld1010 = "[ Reports ]";
}
else
{
    $fld7 = "";
    $fld10 = "";
    $fld107 = "";
    $fld1010 = "";
}
if ($fldUserRights>1) //waiter or chef
{
    $fld11 = "MakeSchedules.php";
    $fld12 = "CheckSchedules.php";
    $fld13 = "ReportSchedules.php";
    $fld1011 = "[Make Schedules]";
    $fld1012 = "[Check Schedules]";
    $fld1013 = "[Report Schedules]";
}
else
{
    $fld11 = "";
    $fld12 = "";
    $fld13 = "";
    $fld1011 = "";
    $fld1012 = "";
    $fld1013 = "";
}
$tpl->set_var("tophr", $fldtophr);

```

```

$tpl->set_var("Field2", $fldField2);
$tpl->set_var("Home", $fldHome);
$tpl->set_var("Menu", $fldMenu);
$tpl->set_var("Reg", $fldReg);
$tpl->set_var("Shop", $fldShop);
$tpl->set_var("Field1", $fldField1);
$tpl->set_var("bottomhr", $fldbottomhr);
$tpl->set_var("fld1", tohtml($fld1));
$tpl->set_var("fld2", tohtml($fld2));
$tpl->set_var("fld3", tohtml($fld3));
$tpl->set_var("fld4", tohtml($fld4));
$tpl->set_var("fld5", tohtml($fld5));
$tpl->set_var("fld6", tohtml($fld6));
$tpl->set_var("fld7", tohtml($fld7));
$tpl->set_var("fld8", tohtml($fld8));
$tpl->set_var("fld9", tohtml($fld9));
$tpl->set_var("fld10", tohtml($fld10));
$tpl->set_var("fld11", tohtml($fld11));
$tpl->set_var("fld12", tohtml($fld12));
$tpl->set_var("fld13", tohtml($fld13));
$tpl->set_var("fld14", tohtml($fld14));
$tpl->set_var("fld15", tohtml($fld15));
$tpl->set_var("fld16", tohtml($fld16));
$tpl->set_var("fld101", tohtml($fld101));
$tpl->set_var("fld102", tohtml($fld102));
$tpl->set_var("fld103", tohtml($fld103));
$tpl->set_var("fld104", tohtml($fld104));
$tpl->set_var("fld105", tohtml($fld105));
$tpl->set_var("fld106", tohtml($fld106));
$tpl->set_var("fld107", tohtml($fld107));
$tpl->set_var("fld108", tohtml($fld108));
$tpl->set_var("fld109", tohtml($fld109));
$tpl->set_var("fld1010", tohtml($fld1010));
$tpl->set_var("fld1011", tohtml($fld1011));
$tpl->set_var("fld1012", tohtml($fld1012));
$tpl->set_var("fld1013", tohtml($fld1013));
$tpl->set_var("fld1014", tohtml($fld1014));
$tpl->set_var("fld1015", tohtml($fld1015));
$tpl->set_var("fld1016", tohtml($fld1016));
$tpl->parse("FormMenu", false);
}
?>

```

```

<?php
/*****
*      Filename: Footer.php
*****/
function Footer_show()
{
    global $tpl;
    // Set URLs
    $fldField1 = "Default.php";
    $fldField3 = "Registration.php";
    $fldField5 = "ShoppingCart.php";
    $fldField2 = "Login.php";
    $fldField6 = "Menu.php";
    // Show fields
    $tpl->set_var("Field1", $fldField1);
    $tpl->set_var("Field3", $fldField3);
    $tpl->set_var("Field5", $fldField5);
    $tpl->set_var("Field2", $fldField2);
    $tpl->set_var("Field6", $fldField6);
    $tpl->parse("FormFooter", false);
}
?>

```

```

<?php
/*****
*      Filename: function.php
*****/
function LB($value, $value2, $value3, $value4, $value5, $value6)
{
    global $tpl;
    $tpl->set_var("$value", "");
    $value2 = new DB_Sql();
    $value2->Database = DATABASE_NAME;
    $value2->User      = DATABASE_USER;
    $value2->Password  = DATABASE_PASSWORD;
    $value2->Host      = DATABASE_HOST;

    $value2->query("select $value3, $value6 from $value5 order by 2");
    while ($value2->next_record())
    {
        $tpl->set_var("ID", $value2->f(0));
        $tpl->set_var("Value", $value2->f(1));
        if ($value2->f(0) == $value4)
            $tpl->set_var("Selected", "SELECTED" );
        else
            $tpl->set_var("Selected", "");
        $tpl->parse("$value", true);
    }
}

function LB1($value, $value2, $value3, $value4, $value5, $value6)
{
    global $tpl;
    $tpl->set_var("$value", "");
    $value2 = new DB_Sql();
    $value2->Database = DATABASE_NAME;
    $value2->User      = DATABASE_USER;
    $value2->Password  = DATABASE_PASSWORD;
    $value2->Host      = DATABASE_HOST;

    $value2->query("select $value3, $value6 from $value5 order by 2
DESC");
    while ($value2->next_record())
    {
        $tpl->set_var("ID", $value2->f(0));
        $tpl->set_var("Value", $value2->f(1));
        if ($value2->f(0) == $value4)
            $tpl->set_var("Selected", "SELECTED" );
        else
            $tpl->set_var("Selected", "");
        $tpl->parse("$value", true);
    }
}

function LB_non($value, $value2, $value3, $value4, $value5, $value6)
{
    global $tpl;
    $tpl->set_var("$value", "");
    $tpl->set_var("ID", "");
    $tpl->set_var("Value", "");
}

```

```

$tpl->parse("$value", true);
$value2 = new DB_Sql();
$value2->Database = DATABASE_NAME;
$value2->User      = DATABASE_USER;
$value2->Password  = DATABASE_PASSWORD;
$value2->Host      = DATABASE_HOST;

$value2->query("select $value3, $value6 from $value5 order by 2");
while ($value2->next_record())
{
    $tpl->set_var("ID", $value2->f(0));
    $tpl->set_var("Value", $value2->f(1));
    if ($value2->f(0) == $value4)
        $tpl->set_var("Selected", "SELECTED" );
    else
        $tpl->set_var("Selected", "");
    $tpl->parse("$value", true);
}
}

function LB_non1($value, $value2, $value3, $value4, $value5, $value6)
{
    global $tpl;
    $tpl->set_var("$value", "");
    $tpl->set_var("ID", "");
    $tpl->set_var("Value", "");
    $tpl->parse("$value", true);
    $value2 = new DB_Sql();
    $value2->Database = DATABASE_NAME;
    $value2->User      = DATABASE_USER;
    $value2->Password  = DATABASE_PASSWORD;
    $value2->Host      = DATABASE_HOST;

    $value2->query("select $value3, $value6 from $value5 order by 2");
    while ($value2->next_record())
    {
        $tpl->set_var("ID", $value2->f(0));
        $tpl->set_var("Value", $value2->f(1));
        if ($value2->f(0) == $value4)
            $tpl->set_var("Selected", "SELECTED" );
        else
            $tpl->set_var("Selected", "");
        $tpl->parse("$value", true);
    }
}

function LB_non2($value, $value2, $value3, $value4, $value5, $value6)
{
    global $tpl;
    $tpl->set_var("$value", "");
    $tpl->set_var("ID", "");
    $tpl->set_var("Value", "All");
    $tpl->parse("$value", true);
    $value2 = new DB_Sql();
    $value2->Database = DATABASE_NAME;
    $value2->User      = DATABASE_USER;
    $value2->Password  = DATABASE_PASSWORD;

```



```

$value2->Host      = DATABASE_HOST;

$value2->query("select $value3, $value6 from $value5 order by 2");
while ($value2->next_record())
{
    $tpl->set_var("ID", $value2->f(0));
    $tpl->set_var("Value", $value2->f(1));
    if ($value2->f(0) == $value4)
        $tpl->set_var("Selected", "SELECTED" );
    else
        $tpl->set_var("Selected", "");
    $tpl->parse("$value", true);
}
}

function LB99($value, $value2, $value3, $value4, $value5, $value6)
{
    global $tpl;
    $tpl->set_var("$value", "");
    $value2 = new DB_Sql();
    $value2->Database = DATABASE_NAME;
    $value2->User      = DATABASE_USER;
    $value2->Password = DATABASE_PASSWORD;
    $value2->Host      = DATABASE_HOST;

    $value2->query("select $value3, $value6 from $value5 order by
category_id");
    while ($value2->next_record())
    {
        $tpl->set_var("ID", $value2->f(0));
        $tpl->set_var("Value", $value2->f(1));
        if ($value2->f(0) == $value4)
            $tpl->set_var("Selected", "SELECTED" );
        else
            $tpl->set_var("Selected", "");
        $tpl->parse("$value", true);
    }
}
?>

```

```

<?php
/*****
*      Filename: Login.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");

session_start();

$filename = "Login.php";

$tpl = new Template($app_path);
$tpl->load_file("Login.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");
$tpl->set_var("FileName", $filename);

$loginErr = "";
$action = get_param("FormAction");
$form = get_param("FormName");
switch ($form)
{
    case "Login":
        login_action($action);
        break;
}

Menu_show();
Footer_show();
Login_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function login_action($action)
{
    global $db;
    global $tpl;
    global $loginErr;

    switch(strtolower($action))
    {
        case "login":          // Login action
            $login = get_param("Login");
            $password = get_param("Password");
            $db->query("SELECT m.member_id, member_level FROM members m, " .
"login l WHERE m.create_time=l.create_time and member_login=" .
            tosql($login, "Text") . " AND member_password=" .
            tosql($password, "Text"));

            if ($db->next_record())
            {
                // Login and password passed
                set_session("UserID", $db->f("member_id"));
                set_session("UserRights", $db->f("member_level"));
            }
        }
    }
}

```

```

        $sPage = get_param("ret_page");
        if (strlen($sPage))
            header("Location: " . $sPage);
        else
            header("Location: ShoppingCart.php");
    }
    else
        $sLoginErr = "Login or Password is incorrect.";
    $tpl->parse("FormLogin", false);
    break;
case "logout": // Logout action
    session_unregister("UserID");
    session_unregister("UserRights");
    $tpl->parse("FormLogin", false);
    break;
}
}

function Login_show()
{
    global $tpl;
    global $sLoginErr;
    global $db;

    $tpl->set_var("sLoginErr", $sLoginErr);
    $tpl->set_var("querystring", get_param("querystring"));
    $tpl->set_var("ret_page", get_param("ret_page"));

    if (get_session("UserID") == "")
    {
        // User did not login
        $tpl->set_var("LogoutAct", "");
        $tpl->set_var("UserInd", "");
        $tpl->set_var("Login", strip(tohtml(get_param("Login"))));
        if ($sLoginErr == "")
            $tpl->set_var("LoginError", "");
        else
        {
            $tpl->set_var("sLoginErr", $sLoginErr);
            $tpl->parse("LoginError", false);
        }
        $tpl->parse("LoginAct", false);
    }
    else
    {
        // User logged in
        $db->query("SELECT member_login FROM login l, members m WHERE " .
            "m.creat_time=l.creat_time and m.member_id=get_session('UserID')");
        $db->next_record();
        $tpl->set_var("LoginError", "");
        $tpl->set_var("LoginAct", "");
        $tpl->set_var("UserID", $db->f("member_login"));
        $tpl->parse("UserInd", false);
    }
    $tpl->parse("FormLogin", false);
}
?>

```

```

<?php
/*****
*      Filename: SubMenu.php
*****/
include ("./common.php");
include ("./Header.php");
include ("./Footer.php");
include ("./function.php");

session_start();

$filename = "SubMenu.php";

$tpl = new Template($app_path);
$tpl->load_file("SubMenu.html", "main");
$tpl->load_file($header_filename, "Header");
$tpl->load_file($footer_filename, "Footer");

$tpl->set_var("FileName", $filename);

Menu_show();
Footer_show();
Results_show();
Search_show();
AdvMenu_show();
Total_show();

$tpl->parse("Header", false);
$tpl->parse("Footer", false);
$tpl->pparse("main", false);

/*****
function Results_show()
{
    global $tpl;
    global $db;
    global $sResultsErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;

    $tpl->set_var("TransitParams", "category_id=" .
tourl(strip(get_param("category_id"))) .
    "&name=" . tourl(strip(get_param("name"))) . "&pricemin="
tourl(strip(get_param("pricemin"))) . "&pricemax=" .
    tourl(strip(get_param("pricemax"))) . "&author=" .
    tourl(strip(get_param("author"))) . "&");
    $tpl->set_var("FormParams", "category_id=" .
tourl(strip(get_param("category_id"))) .
    "&name=" . tourl(strip(get_param("name"))) . "&pricemin=" .
    tourl(strip(get_param("pricemin"))) . "&pricemax=" .
    tourl(strip(get_param("pricemax"))) . "&author=" .
    tourl(strip(get_param("author"))) . "&");
    // Build WHERE statement
    $pcategory_id = get_param("category_id");
    if (is_number($pcategory_id) && strlen($pcategory_id))

```

```

    $pcategory_id = round($pcategory_id);
else
    $pcategory_id = "";
if (strlen($pcategory_id))
{
    if ($sWhere != "") $sWhere .= " and ";
    $HasParam = true;
    $sWhere .= "i.category_id=" . $pcategory_id;
}
$name = get_param("name");
if (strlen($name))
{
    if ($sWhere != "") $sWhere .= " and ";
    $HasParam = true;
    $sWhere .= "i.name like " . tosql("%".$name."%", "Text");
}
$ppricemax = get_param("pricemax");
if (is_number($ppricemax) && strlen($ppricemax))
    $ppricemax = round($ppricemax);
else
    $ppricemax = "";
if (strlen($ppricemax))
{
    if ($sWhere != "") $sWhere .= " and ";
    $HasParam = true;
    $sWhere .= "i.price<" . $ppricemax;
}
$ppricemin = get_param("pricemin");
if (is_number($ppricemin) && strlen($ppricemin))
    $ppricemin = round($ppricemin);
else
    $ppricemin = "";
if (strlen($ppricemin))
{
    if ($sWhere != "") $sWhere .= " and ";
    $HasParam = true;
    $sWhere .= "i.price>" . $ppricemin;
}
if ($HasParam)
    $sWhere = " AND (" . $sWhere . ")";

$sDirection = "";
$sSortParams = "";

// Build ORDER statement
$sOrder = " order by i.name Asc";
$iSort = get_param("FormResults_Sorting");
$iSorted = get_param("FormResults_Sorted");
if (!$iSort)
    $tpl->set_var("Form_Sorting", "");
else
{
    if ($iSort == $iSorted)
    {
        $tpl->set_var("Form_Sorting", "");
        $sDirection = " DESC";
        $sSortParams = "FormResults_Sorting=" . $iSort .

```

```

"&FormResults_Sorted=" . $iSort . "&";
    }
    else
    {
        $tpl->set_var("Form_Sorting", $iSort);
        $sDirection = " ASC";
        $sSortParams = "FormResults_Sorting=" . $iSort .
"&FormResults_Sorted=" . "&";
    }

    if ($iSort == 1) $sOrder = " order by i.name" . $sDirection;
    if ($iSort == 3) $sOrder = " order by i.price" . $sDirection;
    if ($iSort == 4) $sOrder = " order by c.name" . $sDirection;
    if ($iSort == 5) $sOrder = " order by i.image_url" . $sDirection;
}

// Build full SQL statement

$sSQL = "select i.category_id as i_category_id, " .
        "i.image_url as i_image_url, i.item_id as i_item_id, " .
        "i.name as i_name, i.price as i_price, " .
        "c.category_id as c_category_id, c.name as c_name " .
        " from items i, categories c where c.category_id=i.category_id ";

$sSQL .= $sWhere . $sOrder;
$tpl->set_var("SortParams", $sSortParams);

// Execute SQL statement
$db->query($sSQL);

// Select current page
$iPage = get_param("FormResults_Page");
if (!strlen($iPage)) $iPage = 1;
$RecordsPerPage = 20;
if (($iPage - 1) * $RecordsPerPage != 0)
    $db->seek(($iPage - 1) * $RecordsPerPage);
$iCounter = 0;

if ($db->next_record())
{
    // Show main table based on SQL query
    do
    {
        $fldname = $db->f("i_name");
        $fldprice = number_format($db->f("i_price"), 2);
        $fldcategory_id = $db->f("c_name");
        $fldimage_url = $db->f("i_image_url");
        $fldimage_url="Image";
        $tpl->set_var("name", tohtml($fldname));
        $tpl->set_var("name_URLLink", "DishDetail.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->set_var("price", tohtml($fldprice));
        $tpl->set_var("category_id", tohtml($fldcategory_id));
        $tpl->set_var("image_url", $fldimage_url);
        $tpl->set_var("image_url_URLLink", "DishImage.php");
        $tpl->set_var("Prm_item_id", tourl($db->f("i_item_id")));
        $tpl->parse("DListResults", true);
        $iCounter++;
    }
}

```

```

    }
    while ($iCounter < $RecordsPerPage &&$db->next_record());
}
else
{
    // No Records in DB
    $tpl->set_var("DListResults", "");
    $tpl->parse("ResultsNoRecords", false);
    $tpl->set_var("ResultsScroller", "");
    $tpl->parse("FormResults", false);
    return;
}

// Parse scroller
if (@$db->next_record())
{
    if ($iPage == 1)
        $tpl->set_var("ResultsScrollerPrevSwitch", "_");
    else
    {
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("ResultsScrollerPrevSwitch", "");
    }
    $tpl->set_var("NextPage", ($iPage + 1));
    $tpl->set_var("ResultsScrollerNextSwitch", "");
    $tpl->set_var("ResultsCurrentPage", $iPage);
    $tpl->parse("ResultsScroller", false);
}
else
{
    if ($iPage == 1)
        $tpl->set_var("ResultsScroller", "");
    else
    {
        $tpl->set_var("ResultsScrollerNextSwitch", "_");
        $tpl->set_var("PrevPage", ($iPage - 1));
        $tpl->set_var("ResultsScrollerPrevSwitch", "");
        $tpl->set_var("ResultsCurrentPage", $iPage);
        $tpl->parse("ResultsScroller", false);
    }
}
$tpl->set_var("ResultsNoRecords", "");
$tpl->parse("FormResults", false);
}

function Search_show()
{
    global $db;
    global $tpl;

    $tpl->set_var("ActionPage", "SubMenu.php");

    // Set variables with search parameters
    $fldcategory_id = strip(get_param("category_id"));
    $fldname = strip(get_param("name"));
    // Show fields
    LB_non2(Lbcategory_id, $dbcategory_id, "category_id",

```

```

$fldcategory_id, "categories", "name");

$tpl->set_var("name", tohtml($fldname));
$tpl->parse("FormSearch", false);
}

function AdvMenu_show()
{
    global $tpl;
    // Set URLs
    $fldField1 = "AdvSearch.php";
    // Show fields
    $tpl->set_var("Field1", $fldField1);
    $tpl->parse("FormAdvMenu", false);
}

function Total_show()
{
    global $tpl;
    global $db;
    global $sTotalErr;
    $sWhere = "";
    $sOrder = "";
    $sSQL = "";
    $HasParam = false;
    $tpl->set_var("TransitParams", "");
    $tpl->set_var("FormParams", "");
    $tpl->set_var("FormParams", "category_id=" .
    tourl(strip(get_param("category_id"))) .
    "&name= tourl(strip(get_param("name"))) . "&author=" .
    tourl(strip(get_param("author"))) . "&pricemin=" .
    tourl(strip(get_param("pricemin"))) . "&pricemax=" .
    tourl(strip(get_param("pricemax"))) . "&");
    // Build WHERE statement
    $pcategory_id = get_param("category_id");
    if (is_number($pcategory_id) && strlen($pcategory_id))
        $pcategory_id = round($pcategory_id);
    else
        $pcategory_id = "";
    if (strlen($pcategory_id))
    {
        if ($sWhere != "") $sWhere .= " and ";
        $HasParam = true;
        $sWhere .= "i.category_id=" . $pcategory_id;
    }
    $pname = get_param("name");
    if (strlen($pname))
    {
        if ($sWhere != "") $sWhere .= " and ";
        $HasParam = true;
        $sWhere .= "i.name like " . tosql("%".$pname ."%", "Text");
    }
    $ppricemax = get_param("pricemax");
    if (is_number($ppricemax) && strlen($ppricemax))
        $ppricemax = round($ppricemax);
    else
        $ppricemax = "";
}

```



## REFERENCES

1. Leon Atkinson, Core PHP Programming: Using PHP to Build Dynamic Web Sites, Second Edition. Prentice-Hall, Inc., 2001.
2. David Medinets, PHP3 Programming Browser-Based Applications, McGraw-Hill Companies, Inc., 2000.
3. Ramez Elmasri & Shamkant B. Navathe, Fundamentals of Database Systems, Third Edition, Addison-Wesley. Inc., 2000.
4. Recommended Practice of Software Requirements Specifications, IEEE Std. 830-1998 IEEE.
5. Martin Fowler, UML Distilled: A Brief Guide To The Standard Object Modeling Language, Second Edition. Addison Wesley Longman Inc., 1999.
6. Installing Apache, MySQL, and on Linux,  
[http://www.host.library.com/installing\\_apache\\_mysql\\_php\\_on\\_Linux](http://www.host.library.com/installing_apache_mysql_php_on_Linux).
7. Randy Jay Yarger, George Reese, and Tim King, MySQL & mSQL, O'RELLY & Associates, Inc., 1999.